A METHODOLOGY TO ENHANCE QUANTITATIVE TECHNOLOGY EVALUATION THROUGH EXPLORATION OF EMPLOYMENT CONCEPTS IN ENGAGEMENT ANALYSIS

A Thesis
Presented to
The Academic Faculty

By

Mackenzie Hing Kiang Lau

In Partial Fulfillment
of the Requirements for the Degree
Doctor of Philosophy in the
School of Aerospace Engineering

Georgia Institute of Technology

August 2021

A METHODOLOGY TO ENHANCE QUANTITATIVE TECHNOLOGY EVALUATION THROUGH EXPLORATION OF EMPLOYMENT CONCEPTS IN ENGAGEMENT ANALYSIS

Approved by:

Prof. Dimitri Mavris, Advisor
School of Aerospace Engineering
Georgia Institute of Technology

Prof. Daniel Schrage
School of Aerospace Engineering
Georgia Institute of Technology

Dr. Michael Steffens
School of Aerospace Engineering
Georgia Institute of Technology

Dr. Kelly Griendling
School of Aerospace Engineering
Georgia Institute of Technology

Mr. James Zeh
Air Force Research Laboratory

Date Approved: July 29, 2021

Let us spread truth, for it is more powerful than any atomic weapon.

Zach Weinersmith

For
瑞
珍

ACKNOWLEDGEMENTS

I have many people to thank for their support over the course of this endeavor. My greatest debt is owed to my family, whose encouragement through the years was an invaluable pillar for me to lean on. Mom, Dad, Chelsea, and Jessica – thank you.

I would like to thank Dr. Vassilis Syrmos of the University of Hawai‘i at Mānoa, whose advocacy and enthusiasm gave me the confidence to begin this journey.

Many thanks are owed to my friends and colleagues at ASDL for the late nights spent grinding out projects, the lazy floats down the Chattahoochee, and everything in between. To Adam Siegel, for helping me find my footing with unparalleled patience and kindness at every step. To Nicky Robertson and Eugina Mendez-Ramos, for welcoming me into the fold and always being there for me. And a special thanks to the denizens of Weber 106 for tolerating my coffee habit, saving me from my own baked goods, and making each day in the lab something to look forward to. It has been a pleasure getting to know you all, and I wish you the best.

I would like to thank the members of my committee: Prof. Daniel Schrage, Dr. Michael Steffens, Dr. Kelly Griendling, and Mr. James Zeh. Your insights and feedback helped me refine my thinking and find clarity when I needed it most. I am especially grateful to Dr. Steffens, whose mentorship always kept me grounded.

Lastly, I am extremely grateful to my advisor, Professor Dimitri Mavris, who gave me a chance and let me decide what to do with it. Your guidance has shaped my thinking and helped me grow in ways I could not have imagined. Thank you for lighting my torch.

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

NOMENCLATURE

Abbreviations

ABM	Agent-Based Model
ABMS	Agent-Based Modeling & Simulation
AFSIM	Advanced Framework for Simulation, Integration, and Modeling
ANN	Artificial Neural Network
AoA	Analysis of alternatives
BE	Beam evasion
CBA	Capabilities-Based Assessment
DoD	United States Department of Defense
DOE	Design of Experiments
DOTmLPF-P	Doctrine, Organization, Training, Materiel, Leadership, Personnel, Facilities, and Policy
DSE	Design Space Exploration
ECWG	Employment Concepts Working Group
FLAME	Flexible Large-scale Agent-based Modeling Environment
FLAMES	Flexible Analysis Modeling and Exercise System
GA	Genetic Algorithm
ICD	Initial Capabilities Document
IPD	Iterated Prisoner's Dilemma
JCIDS	Joint Capabilities Integration and Development System
LHS	Latin Hypercube Sampling
LOS	Line-of-sight

M&S	Modeling & Simulation
MADM	Multi-Attribute Decision Making
MARL	Multi-Agent Reinforcement Learning
MDO	Multidisciplinary Design Optimization
ML	Machine Learning
MOE	Measure of Effectiveness
MOP	Measure of Performance
MTOW	Maximum Takeoff Weight
PD	Prisoner's Dilemma
PE	Pure evasion
PMF	Probability Mass Function
PN	Proportional navigation
PP	Pure pursuit
PPO	Proximal Policy Optimization
PSO	Particle Swarm Optimizer
RL	Reinforcement Learning
SAW	Simple Additive Weighting
SOCRATES	Simulated-based, Object-oriented, Capability-focused, Real-time Analytical Technology Evaluation for Systems-of-systems
TOPSIS	Technique for Order Preference by Similarity to Ideal Solution

SUMMARY

The process of designing a new system has often been treated as a purely technological problem, where the infusion or synthesis of new technologies forms the basis of progress. However, recent trends in design and analysis methodologies have tried to shift away from the narrow scope of technology-centric approaches. One such trend is the increase in analysis scope from the level of an isolated system to that of multiple interacting systems. Analysis under this broader scope allows for the exploration of non-materiel solutions to existing or future problems. Solutions of this type can reduce the cost of closing capability gaps by mitigating the need to procure new systems to achieve desired levels of performance. In particular, innovations in the employment concepts can enhance existing, evolutionary, or revolutionary materiel solutions.

The task of experimenting with non-materiel solutions often falls to operators after the system has been designed and produced. This begs the question as to whether the chosen design adequately accounted for the possibility of innovative employment concepts which operators might discover. Attempts can be made to bring the empirical knowledge possessed by skilled operators upstream in the design process. However, care must be taken to ensure such attempts do not introduce unwanted bias, and there can be significant difficulty in translating human intuition into an appropriate modeling paradigm for analysis. Furthermore, the capacity for human operators to capitalize on the potential benefits of a given technology may be limited or otherwise infeasible in design space explorations where the number of alternatives becomes very large. This is especially relevant to revolutionary concepts to which prior knowledge may not be applicable. Each of these complicating factors is exacerbated by interactions between systems, where changes in the decision-making processes of individual entities can greatly influence outcomes. This necessitates exploration and analysis of employment concepts for all relevant entities, not only that or those to which the technology applies.

This research sought to address the issues of exploring employment concepts in the early phases of the system design process. A characterization of the problem identified several gaps in existing methodologies, particularly with respect to the representation, generation, and evaluation of alternative employment concepts. Relevant theories, including behavioral psychology, control theory, and game theory were identified to facilitate closure of these gaps. However, these theories also introduced technical challenges which had to be overcome. These challenges stemmed from systematic problems such as the curse of dimensionality, temporal credit assignment, and the complexities of entity interactions. A candidate approach was identified through thorough review of available literature: Multi-agent reinforcement learning. Experiments show the proposed approach can be used to generate highly effective models of behavior which could out-perform existing models on a representative problem. It was further shown that models produced by this new method can achieve consistently high levels of performance in competitive scenarios. Additional experimentation demonstrated how incorporation of design variables into the state space allowed models to learn policies which were effective across a continuous design space and outperformed their respective baselines. All of these results were obtained without reliance on prior knowledge, mitigating risks in and enhancing the capabilities of the analysis process. Lastly, the completed methodology was applied to the design of a fighter aircraft for one-on-one, gun-only air combat engagements to demonstrate its efficacy on and applicability to more complex problems.

CHAPTER 1
INTRODUCTION

"To say that something is ordinary is to say it is of the kind that has made the biggest contribution to the formation of your most basic ideas."

— Paul Valéry

Advances in fighter aircraft design have brought with them radical changes to the way the United States Air Force approached its missions and conducted its operations. Its design emphasized several advanced technologies, such as low-observability, network-centric warfare, data enrichment through sensor fusion, and supersonic cruise. These novel capabilities were believed to be essential to mission success in future operations of the United States and its allies.

A key challenge in designing a system with novel technologies lies in developing an understanding for how those technologies will be employed. Test pilots with the US Marine Corps were "pushing themselves to push past planned tactics and create a new way of using the fifth-generation technology" [39]. Pilots were actively discovering new, unplanned, and innovative tactics for employing, leveraging, and enhancing the capabilities offered by the advanced technologies.

1.1 A Brief History of Tactical Innovation

Tactical innovation has been a staple of the US military. During World War II, US Navy pilot John Thach developed his eponymous weaving maneuver as a response to the capability differences between his Grumman F4F and the Mitsubishi A6M flown by his adversaries. The Beam Defense Maneuver, shown in Figure 1.1, enabled a pair of pilots to work together and stay in the fight against a more agile opponent. This maneuver was developed by Thach over the course of many late nights as he toyed with matchsticks as stand-ins for

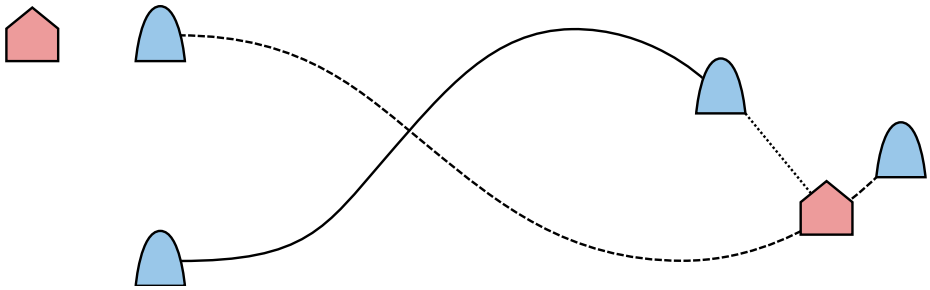


Figure 1.1: The Beam Defense Maneuver

engaging aircraft [142]. Test flights demonstrated the potential for this novel tactic, which altered the number of aircraft in a section and reduced the reliance on radio communications for coordination. Ultimately, the efficacy of this novel tactic is impossible to separate from the myriad other factors which influenced the course of the war.

Several major advancements in fighter aircraft design occurred near the conclusion of World War II. First, jet propulsion became increasingly feasible [121]. Jet-powered aircraft could fly significantly faster than their propeller-driven predecessors, and this brought with it a change in the way air combat was conducted. Radar technology also improved dramatically, enhancing situational awareness and altering the conduct of operations [66]. The ability to detect threats from longer ranges allowed for better planning and coordination of engagements, especially when those threats were fast-moving, jet-powered aircraft. Lastly, the development of missile systems altered the envelope within which enemies could be engaged compared to guns and cannons [134, 123].

The F-86 was one of the first jet-powered aircraft to see extensive usage by the US [93]. Its adversary was the Mikoyan-Gurevich MiG-15, which was visually similar and comparable in terms of thrust, but came in at three-fourths the weight of the F-86. This gave the MiG-15 an advantage in terms of climb rate and service ceiling. However, the F-86 had a slight edge in terms of fire rate, was more capable in a dive, and used advanced systems to enhance responsiveness of control surfaces at high speeds [19]. There was also a discrepancy in terms of weaponry. The MiG-15 carried three guns – one 37-mm and two 23-mm – to conduct intercept operations against bomber aircraft. By comparison, the F-86

carried six guns, each using .50-cal. or 12.7-mm round [93]. The guns on the F-86 packed less of a punch but it could put more rounds downrange in a given amount of time. Pilots of the F-86 ultimately dominated their adversaries flying MiG-15s, not only because of their technological advantages but also their experience and capabilities as operators [152].

Jet propulsion, radar, and missiles were all present in the design of the McDonnell Douglas F-4 Phantom II. However, these advanced technologies may have given designers a misguided sense of security [43]. A belief at the time was that the availability of radar and missiles had rendered dog fighting obsolete [95, 147]. The F-4 Phantom II was initially designed without a gun for this reason. This design choice saved weight by omitting the weapon system and associated ammunition, and may have improved high speed aerodynamics since the protruding gun would not disrupt air flow around the vehicle. However, it also left the F-4 extremely vulnerable should it be engaged at short ranges where the use of missiles would be dangerous or impractical. Its only defense in a gun fight would have been its ability to accelerate to high speeds, since the aerodynamic requirements for supersonic flight hampered maneuverability at lower speeds [6, 34].

Another blow to the F-4 design was the disappointing reality of missile performance at the time. The "primitive and unreliable" Sidewinder missiles left much to be desired [18]. Ultimately, the F-4 Phantom II was downed at a rate of 0.721 per 1,000 combat sorties over the course of the Vietnam War – nearly double the average over all aircraft flown during the conflict [48]. This was a marked improvement over the loss rates realized in World War II and the Korean War – 9.7 and 2.0, respectively [115] – but indicated the development of new technologies was out of step with the tactical reality of war at the time.

1.1.1 The Motivating Question

Accounts of armed conflicts from the past 80 years have highlighted a persistent gap between technological and tactical advances by the United States Department of Defense (DoD). These observations showed how the two dimensions of military operations could

fall out of sync with one another and established the potential for such gaps to adversely impact mission effectiveness. It would be ideal if the operational experience of operators, as well as their potential innovations in the operational environment, could be captured by the system design process in order to ensure the end product effectively capitalizes on the ways in which a system could be employed in the battlespace. Furthermore, it would be ideal if analysts could gain insights into how potential adversaries might alter their operational concepts in order to mitigate any technological disparities in the future. Synthesis of these considerations lead to the formulation of the motivating question for this research:

Motivating Question

How can explorations of tactics be incorporated into the system design process?

The first step in approaching the motivating question was to establish the context of the problem. The motivating question could be broken into two parts: Explorations of tactics, and system design processes. These concepts will be expounded upon in the following sections, beginning with system design processes. Considerations for how explorations of tactics could be incorporated in those processes will then be addressed.

1.2 A Generic Process for System Design

The process or processes employed in a system design effort can be highly variable, and many attributes of any given process will be problem-specific. Schrage and Mavris established a generic decision-making process to capture the essence of any design effort [116]. The six steps to this process are shown in Figure 1.2. Each step will be examined more closely in the following paragraphs.

The first step in making a decision is to establish the need for the effort. Needs could come from several sources, such as economic factors, environmental considerations, or the impending obsolescence of existing systems. In essence, a need arises when existing solutions have been observed or predicted to fall short of some threshold level of performance.

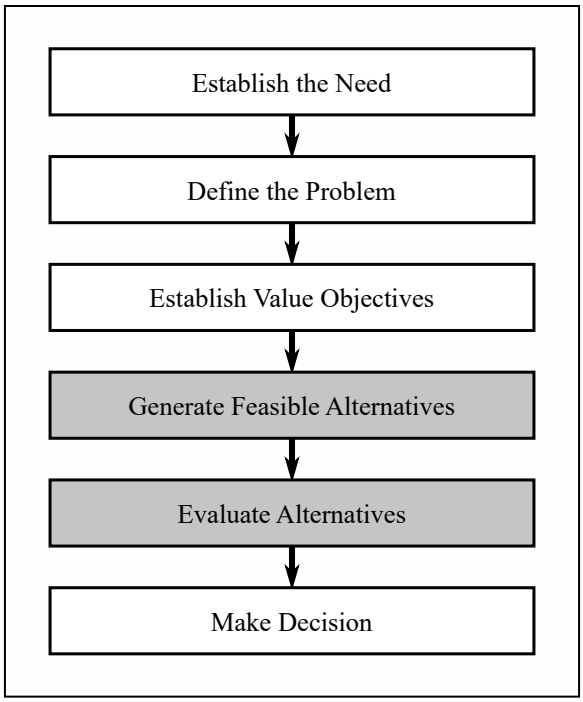


Figure 1.2: A generic decision-making process, adapted from [116]

A problem definition would be required once the need has been established. This would involve defining the context of the need, the kind and degree of observed or predicted shortfalls, and, when possible, the causes of those shortfalls. A clear problem definition can facilitate the remainder of the process.

Establishing the value objectives of the decision-making process "includes establishing feasibility constraints and criteria" [117]. Constraints could be imposed on relevant metrics, such as minimum operational range or maximum takeoff weight (MTOW) for an aircraft, which are to be estimated in the subsequent steps. Establishing clear and proper value objectives is critical to effective decision making because they define the boundaries of the design space, constraining those alternatives which would be considered, and provide an order relation for comparing feasible designs. This order relation can be used to assess the relative preference of any design over another based on the stated criteria. For example, one might compare an aircraft with a high MTOW but short range against another with greater range but lower MTOW. If the value objectives are not clearly stated then ef-

fective comparison of these two alternatives might not be possible. However, if the decision maker established, from the outset, that range was far and away the most important factor, and MTOW was secondary, then the second alternative might be distinctly preferable to the first. The exact value of each alternative would depend on the estimated metrics and weightings assigned to them.

If the need has been established, the problem clearly defined, and the value objectives clearly stated then the most intensive steps of the design process can begin: Generating and evaluating feasible alternatives. First, what constitutes a "feasible" alternative must be established. If something is "feasible" then it is "capable of being done" [41] or "able to made, done, or achieved" [42]. A feasible alternative, then, is one which is either currently possible or expected to be possible within the time frame of the project. This precludes outlandish technologies or concepts which might not be sufficiently mature to warrant analysis.

Feasibility serves to constrain the number of alternatives to be evaluated. Such constraints are necessary because design spaces can be extremely large, and reducing their size can help to ensure the effort proceeds logically and efficiently. If the design space is not adequately constrained then resources may be wasted in analyzing alternatives which would not be seriously considered by decision makers.

1.2.1 A Modern Approach to Acquisitions

The DoD introduced the Joint Capabilities Integration and Development System (JCIDS) at the start of the 21st century to guide the design, development, and acquisition of new military systems. Its primary innovations were in the language used and management of information throughout the acquisition process.

The purpose of the new system was to "ensure the capabilities required ... are identified with the associated operational performance criteria in order to successfully execute the missions assigned" [83]. In simpler terms, the DoD wanted to take a more holistic view of its missions. This new view would treat technological advancements as means to ends

rather than ends unto themselves. The ends, then, would be the capabilities possessed by the US military.

JCIDS defines a capability as "the ability to complete a task or execute a course of action under specified conditions and level of performance" [51]. There could exist conditions for which no currently accepted course of action could complete the task to a satisfactory level of performance, and such a circumstance would constitute a capability gap. The purpose of the JCIDS process is to identify and characterize these gaps early in the acquisition process, and to direct further efforts towards closing those gaps.

A Capabilities-Based Assessment (CBA) is "the starting point in identifying the DoD's needs and recommending solutions" [26]. This maps directly to the first step in the generic decision-making process. According to JCIDS:

> The CBA identifies: the capabilities and operational performance criteria required to successfully execute missions; the shortfalls in existing weapon systems to deliver those capabilities and associated operational risks; the possible non-materiel approaches for mitigating or eliminating the shortfall, and when appropriate recommends pursuing a material solution. – *CJCSI 3170.01G*

The purpose of a CBA is primarily to describe, identify, and justify the need for addressing capability gaps. In this way, a CBA encapsulates the first three steps of decision-making. The DoD may decide to move forward on developing new capabilities if the need to mitigate the operational risks associated with the identified gap is deemed sufficient.

Closing Capability Gaps

The output of CBA is the Initial Capabilities Document (ICD) [26]. There are seven major elements of an ICD, including a concept of operations, list of functional areas, description of capabilities, description of gaps and associated metrics, summary of threats and environments, proposals for possible solutions, and a set of final recommendations. JCIDS emphasizes that a CBA should not include detailed analysis or specification of a solution [26].

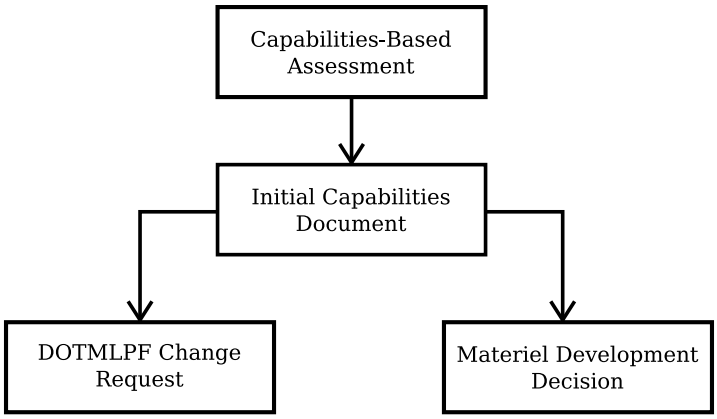


Figure 1.3: CBA in the DoD needs identification process. Adapted from Figure 1-2 in [26]

Instead, its goal should be to provide information in support of a decision as to what type of solution to pursue. Solutions are divided into two categories: Materiel and non-materiel. The solution process forks at this point, as shown in Figure 1.3. The two branches of the solution process will be examined in the following paragraphs.

Non-Materiel Solutions

There are eight non-materiel approaches identified for addressing identified capability gaps, given in Table 1.1. Together, these approaches are known as DOTmLPF-P , which stands for doctrine, organization, training, materiel, leadership, personnel, facilities, and policy. JCIDS documentation states that, "where a new non-materiel capability solution is [desired,] ... one or more aspects of DOTmLPF-P may be changed" [51]. The inclusion of "materiel" in the non-materiel class of solutions may seem paradoxical, but JCIDS draws a distinction between the use of *existing* and the acquisition of *new* materiel [51]. The materiel included in DOTmLPF-P are those which already exist and may be repurposed or otherwise leveraged in order to solve the problem.

Materiel Solutions

CBA may provide some recommendations as to the form of a solution, particularly for materiel solutions. There are four types of solutions a CBA might recommend. The first

Table 1.1: Eight non-materiel approaches to closing capability gaps

Doctrine	Guiding principles regarding the employment and coordination of assets to achieve a common goal
Organization	The hierarchy of command and structure of cooperation within an operational unit
Training	Rehearsal of missions using established doctrine to facilitate mission effectiveness
materiel	Equipment necessary to operate, maintain, and support activities
Leadership	Complement to training, encompassing experience, education, and professional development
Personnel	Qualified persons supporting operations and performing missions
Facilities	Installations and other properties which support operations or programs
Policy	Any intranational or international policies which impact operations

is a recapitalization solution to leverage existing systems, such as restarting or augmenting production of materiel which is already in use. The recapitalization of the Lockheed C-5 Galaxy line in the 1980s exemplifies the utility of recapitalization solutions [99]. Flaws in the material composition of the wing structures limited the cargo capacity of the original C-5A produced in the 1960s and 1970s. The wing was redesigned with new wings in the 1980s to resolve these issues and return the system to its full capacity.

Recapitalization solutions are not always feasible because threats are prone to change over time. Variation in the threat environment might necessitate variation in the solutions employed. Advances in technological capabilities might also warrant their infusion into existing systems to augment or enhance current capabilities. These solutions are evolutionary in nature, generally taking the form of updates or upgrades to existing systems. The McDonnell Douglas F/A-18 Hornet is a prime example of evolutionary solutions in action. The first variants of the Hornet – the F/A-18A/B – entered service in the early 1980s as

all-weather, carrier-capable tactical fighters serving attack, counter air, and reconnaissance roles [150]. A second generation of Hornets – the F/A-18C/D – entered service in the late 1980s and incorporated "provisions for employing updated missiles and jamming devices against enemy ordnance" [155]. The third generation of Hornets – the F/A-18E/F Super Hornet – saw several dramatic changes made to the basic system [150]. The vehicle was lengthened, while its wingspan and height were reduced. The planform of the vehicle was modified to include leading edge root extensions, which improved maneuverability at high angles of attack [53]. The twin General Electric F404s were replaced by twin F414s, producing an extra 4,600 pounds of static thrust total. MTOW was increased by nearly 15,000 pounds (27%) without sacrificing combat range, which increased by nearly 200 nautical miles (17%). These advances were enabled through the infusion of technological advances into the airframe as the former became available, and allowed capabilities to evolve alongside the operating environment.

Transformational solutions are the third type which a CBA might recommend. These types of solutions are radical departures from the status quo. Examples of transformational solutions include jet propulsion, missiles, low-observability, and unmanned systems. Each of these dramatically altered the way military operations were conducted by introducing radically different capabilities into the operating environment. The General Atomics MQ-1 Predator is an example of a transformational solution. It was designed primarily as a remotely piloted reconnaissance platform for medium-altitude, long-endurance operations, and was later adapted to fill the role of armed reconnaissance [27]. It provided "persistent intelligence, surveillance and reconnaissance information combined with a kill capability to the warfighter" [122]. The ability to loiter on station for up to 40 hours coupled with multimode sensors, advanced communications capabilities, and an efficient, low-speed aerodynamic design altered how the airborne assets operated, whether in support of ground forces or as standalone strike platforms [132, 148].

Defining the Scope of This Research

Non-materiel candidates are often the first to be explored in attempting to close a capability gap [109]. However, these types of solutions may not be enough to meet the value objectives and close the capability gaps, especially when threats and/or operational environments are changing in significant ways. An example of this was the development of radio detection and ranging (radar) capabilities during WWII. Early detection using radar allowed the German air forces to organize a defense against inbound Allied bombers [56].

Further advances in radar technology after WWII drove a continuing need for capabilities above and beyond what had existed until then. Countermeasures, such as chaff and jamming, were developed to mitigate the advantages afforded by radar systems but the character of the operational environment had been permanently altered [66]. However, technology would advance to a point where "radar's ability to guide each phase of the [engagement] would threaten to curtail the ability of aircraft to control the skies over the battle space" [56]. Ultimately, "the [United States Army Air Force] learned that air power meant ... harnessing technology and science to produce new [materiel] ... that ensured success in combat" [84]. That is, non-materiel solutions were insufficient to meet the needs of the force as it faced changing threats and new materiel was needed to ensure effectiveness.

The methods employed for identifying non-materiel solutions to capability gaps are present in the analysis of materiel solutions. Doctrine, training, leadership, and policy would be likely to change in response to the introduction of new materiel. It has been acknowledged that past efforts to acquire new materiel were "set up to deal with ... evolutionary improvements in military capability, operating within the well-defined context of existing doctrine" [63]. Further, there are several transformational technologies which are currently in development and expected to exert influence over the operational environment for decades to come [100]. Taken together, these observations indicate the analysis of new materiel solutions, specifically evolutionary and transformational ones, is where the largest gap in the system design process exists. These considerations are summarized in the state-

ment of the research context below.

The Research Context

This work focused on exploring tactics to enhance the design process in support of the analysis of evolutionary and transformational solutions to capability gaps in the form of new materiel

1.2.2 Analysis of Alternatives

A CBA will identify capability gaps, establish the need for addressing those gaps, and establish the associated value objectives which must be met in order to close the gaps. It will also identify the forms of candidate materiel and non-materiel solutions to those gaps. The next steps in the design process would be to generate and evaluate feasible alternatives from those candidate solutions. JCIDS combined these steps into a single process: The analysis of alternatives (AoA).

Generating Alternatives

An AoA is typically focused on comparing alternative recapitalization, evolutionary, and/or transformational solutions. The intent is to "help decision-makers understand the tradespace ... to satisfy an operational capability need" [98]. This work was primarily concerned with evolutionary and transformational solutions since the other types of solutions do not involve any form of system design.

The tradespace must be identified before any analyses can be conducted. In general, the tradespace will consist of the characteristics of the system which can be manipulated to facilitate mission performance, effectiveness, and success. Identifying the tradespace involves establishing constraints on system characteristics, such as the maximum empty weight of an aircraft or a minimum payload capacity.

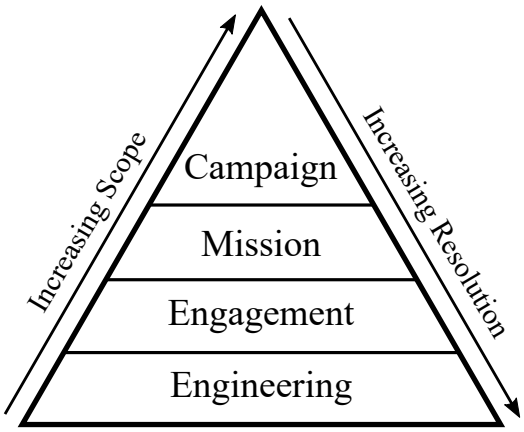


Figure 1.4: Analysis hierarchy, adapted from [97]

There may be a large number and wide variety of characteristics which are available for manipulation in AoA. This would be especially relevant to transformational systems where fewer constraints might be imposed by existing processes. However, even evolutionary solutions can have large design spaces. An even larger problem would arise if the analysts sought to compare both evolutionary and transformational systems. That is, if the question were: Do we need a transformational solution, or can we employ an evolutionary one?

Levels of Analysis

Analyses can take place across several levels of abstraction, as shown in Figure 1.4 [97]. At the bottom of the diagram are the engineering analyses, which have the narrowest scope and highest resolution. Analyses at these levels are often conducted on the components of a system, such as aircraft engines or control surfaces. Analyses at the engagement level consider the interactions between systems. These interactions may take the form of one-on-one or many-on-many combat scenarios. Mission- and campaign-level analyses are even more abstract, and may consider force-on-force scenarios or long-term strategic goals.

An important aspect of the analysis hierarchy is the flow of information between levels. For example, the analysis of alternatives at the engagement level will depend on information obtained through engineering-level analyses. Furthermore, considerations for mission- and campaign-level analyses will inform and be informed by engagement-level results. In

this way, the analysis hierarchy is a structure of information flows, and this emphasizes the importance of ensuring the information obtained at each level adequately captures potential realities. If alternatives at any level are not adequately explored then there may be unintended or unforeseen consequences at the other levels of analysis.

The focus of an AoA is to provide an "analytical rationale for the selection of the best solution in terms of cost and operational effectiveness to support a program decision" [125]. Operational effectiveness relates to the value objectives established early on in the decision-making process. A technology-focused design process would reside almost entirely within the engineering level of analyses, but the *effects* of those technologies on operational effectiveness might only be realized at higher levels. For example, the AoA handbook identifies probability of survival as a potential measure used in the decision-making process, where the value objective is a probability no less than 85% [98]. Engineering-level solutions – i.e. technologies – could be employed to enhance survivability, such as self-sealing fuel tanks or adaptive control systems [12]. However, the effects these technologies have on survivability might not be immediately apparent when analyses are confined to the engineering level. Instead, engagement-level analyses might be needed in order to understand how those technologies influence operational effectiveness.

The effects of engineering-level analyses at the mission or campaign level might be obfuscated by the broadened scope and lowered resolution employed. Analyses would likely have to pass through the engagement level first in order to determine whether or not a technology allows the system of interest to meet the value objectives, e.g. whether or not the use of self-sealing fuel tanks realizes a probability of survivability greater than 85%. The scope of this research was further constrained to the engagement level of analysis for this reason.

The Role of Employment Concepts

Several factors must be considered in the course of a study plan for AoA. One key factor is Employment Concepts associated with the identified capability gap and any candidate solutions [98]. A working group may be tasked with identifying and developing employment concepts to support the evaluation of materiel alternatives.

The Employment Concepts Working Group (ECWG) is tasked with researching existing employment concepts for the baseline and alternatives. Missions and tasks performed must be considered, along with requisite skill-sets for executing operations. The ECWG is responsible for identifying and developing the tactics, techniques, procedures, and doctrine which would be utilized in the analysis of materiel solutions.

JCIDS defines doctrine as the "guiding principles regarding the employment and coordination of assets to achieve a common goal" [51]. The Oxford English Dictionary defines a principle as "a rule or belief governing one's behavior" [106]. That is, principles are the driving forces behind behaviors.

The definition of a behavior is more difficult to ascertain. According to Skinner, "behavior has that kind of complexity or intricacy which discourages simple description and in which magical explanatory concepts flourish abundantly" [130]. He held the position that behaviors were not to be understood, let alone explained. Behaviors may only be observed, perhaps quantified, and sometimes influenced. Even today, a precise definition of behavior remains elusive [38]. A number of definitions have been proposed, despite any perceived or real limitations on our capacity to explain behaviors:

> Observable activity of an organism; anything an organism does that involves action and/or response to stimulation – *Wallace et al. 1991*
>
> The way an organism responds to stimulation – *Raven and Johnson 1989*
>
> A response to external and internal stimuli – *Starr and Taggart 1992*

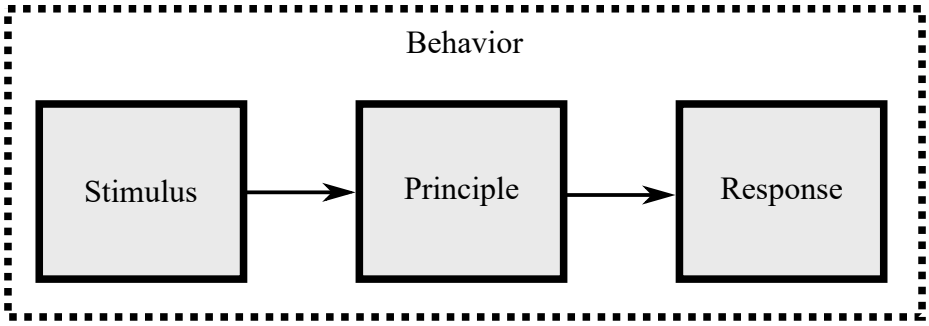


Figure 1.5: A view of behaviors as the mapping of stimuli to responses

> Applied behavioral research is concerned with the manipulation of environmental stimuli to help individuals efficiently and effectively emit specific responses – *Cooper 1982*

There are important commonalities in these definitions. First, that behaviors are attributable to organisms. An organism may be defined as "a complex structure of interdependent and subordinate elements whose relationships and properties are largely determined by their function in the whole" [101]. This conceptualization and definition of an organism is very similar to that of a system:

> A system is a collection of entities and their interrelationships gathered together to form a whole greater than the sum of the parts – *Boardman 2006*

If the concepts of systems and organisms are viewed as interchangeable then the definitions of behaviors as they pertain to organisms may be extended to systems.

The second important commonality among the aforementioned definitions of behavior is the idea that behaviors are responses to stimuli. This is important because it facilitates the investigation of behaviors as functions or mappings from a space of possible stimuli to a space of possible responses. Furthermore, this notion may be brought to bear on the definition of doctrine, specifically with regard to the "guiding principles". Adopting this view would establish principles as the rules, both implicit and explicit, which enable an organism to exhibit a response to a given stimulus, as depicted graphically in Figure 1.5.

Behaviors in the Analysis of Alternatives

The task of ECWG is to identify the principles of behavior which should be employed for analysis of the baseline and alternative materiel solutions to capability gaps. This poses a significant challenge to the analysis of transformational solutions because "there will be no textbook answer on the best use of something totally new, and merely plugging the innovation into an existing [Concept of Operations] probably won't work" [26]. Transformational solutions will require tactical innovation. Recapitalization and evolutionary solutions may also require innovation in order fully close a capability gap. The task of the ECWG is to seek out these innovative tactics and propose them for further analysis.

Identifying the combination of technologies and tactics which maximize the expected performance of a given system would depend on two characteristics of the design process: knowledge and freedom. Design knowledge relates to the processes by which one may consider the consequences and implications of alternative decisions. Design freedom refers to the number of alternatives available, as well as the capacity to consider each.

The trade-off between knowledge and freedom as they pertain to a system design process was examined by Mavris et al. [82]. They identified general trends in knowledge, freedom, and committed cost, which are shown in Figure 1.6. Design freedom is highest early in the process, when few constraints have been imposed and the number of possible alternative paths remains high. However, this high freedom comes at the cost of low knowledge since there are so many possibilities that making comparisons between them becomes impractical, if not impossible. Designs can impose constraints, such as the scenarios of interest or subsets of alternatives which will be examined based on technology maturity, to reduce the size of the design space, trading design knowledge for design freedom. The exchange rate between design knowledge and design freedom is often not one-to-one, since large gaps in knowledge could remain after significant freedom has been sacrificed.

There exists a similar trade from the perspective of tactical employment of a system. The number of possible alternative tactics for any given combination of design and scenario

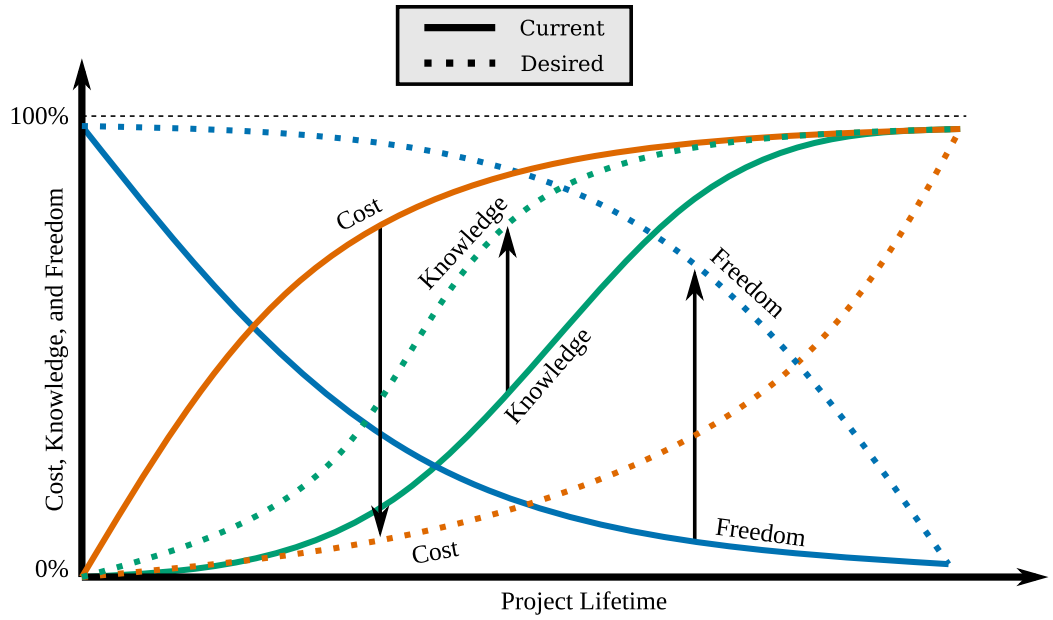


Figure 1.6: Notional trends for cost, knowledge, and freedom versus project lifetime, adapted from [82]

starts out very high, corresponding to a high degree of freedom. However, there would also be very little knowledge about how those alternatives compare and making such comparisons can be difficult. A subset of tactical alternatives may be selected for further analysis, facilitating the acquisition of relevant knowledge at the expense of freedom. In this way, the process of "designing" tactics closely resembles that of designing new systems.

1.3 Innovation Through Experimentation

The process of identifying employment concepts to accompany materiel alternatives to closing capabilities gaps is fundamentally one of experimentation. The term derives from the Latin word expiriri, meaning "to try" [2]. An experiment can be defined as "an operation or procedure carried out under controlled conditions in order to discover an unknown effect" [40]. This type of experimentation is known as "discovery experimentation". Exercise Red Flag fits this definition of an experiment very well. Alberts related discovery experimentation to the military acquisition process directly.

Discovery experiments are similar to the time honored military practice by

> which new military hardware (aircraft, tanks, etc.) was developed against a set of technical specifications (fly faster, turn tighter, shoot farther, etc.), then given to technical user communities (typically Service test organizations or boards) to work out the concepts of operation, tactics, techniques, and procedure for effective employment. – *Alberts 2002 [2]*

1.3.1 Risks in Experimentation

Alberts and Hayes identified five risks associated with the conduct of experiments which they argued should be mitigated to the greatest extent possible. These will be collectively referred to as the "Five Risks" in the remainder of this document for the sake of brevity. The first risk is "moving ahead without sufficient evidence and understanding" [3]. Alberts argues this risk pertains to finding the appropriate balance between conducting too few experiments and too many [2]. A single experiment often cannot prove the existence or degree of a relationship between phenomena. On the other hand, the volume of data produced through excessive experimentation can hinder the process by overwhelming the analysts. These considerations are especially relevant to experiments which rely on stochastic elements, where characterizing the distribution of relationships becomes an important aspect of understanding the observed phenomena.

The second risk is "prematurely settling on an approach", which is closely tied to the first risk. Alberts and Hayes relate this risk to "fast tracking" a candidate solution without conducting adequate discovery experimentation. An example of this might be found in the case of missile technology with the F-4. The technology showed promise as a concept: engage at longer ranges and capitalize on other technologies, specifically radar. In hindsight, the new technologies were immature and should not have been as heavily relied upon at the time. A more thorough process of experimentation may have revealed the potential pitfalls of the nascent technology and prevented the need for a redesign of the system later on.

The third risk lies in confining exploration experiments to well-established borders.

This risk is especially pertinent to experimentation with employment concepts for materiel solutions to capability gaps. Innovative tactics lie beyond the boundaries of established knowledge by definition.

Conducting experiments through a process of trial and error is the fourth risk identified by Alberts and Hayes. They argue that experiments of all sorts should be guided by some form of theory, without which progress can be "unsure, inefficient, and relatively slow" [3]. Cognitive science is cited as a relevant scientific discipline. Concepts from behavioral science, such as Skinner's theory of operant conditioning, may also be of use in these types of experiments.

The last risk lies in failing to capitalize on the creativity of the end-user. This risk is firmly rooted in the discovery and analysis of innovative employment concepts. Attempts have been made to capture the human factor in the design of new aircraft, such as when, in 1942, a Grumman engineer was sent to interview John Thach and ascertain what design elements would be desirable from the perspective of the pilot who had advanced the state of art in air combat tactics [159]. However, relying on this form of input may run afoul of the other risks, and care must be taken to strike a proper balance between them.

Thinking Like the Enemy

A critical element to the analysis of military operations is properly characterizing how potential adversaries might respond. John Thach developed his eponymous weaving maneuver by imagining how a Japanese pilot might behave when pursuing a target. F-4 pilots were forced to contend with an enemy who could engage at much shorter ranges than were ideal for the missiles employed. These examples highlight importance of considering how an adversary might behave in an engagement, and how those behaviors might influence the outcome.

The US and its allies acknowledge the importance of "thinking like the enemy" through the conduct of simulated engagements for training purposes. Exercise Red Flag is a notable

example, where several countries simulate military engagements and conduct exercise to maintain readiness where "aerial adversary tactics [are] ... a key focus" [86]. This comes at the cost of around US$3.5 million per exercise [47].

The 57th Adversary Tactics Group plays the role of the enemy in Exercise Red Flag. Aggressor pilots are "specially trained to replicate the tactics and techniques of potential adversaries" [149]. Simulated engagements allow pilots to hone their tactics and express their creativity. In the best case, pilots would be able to explore and learn new behaviors which could capitalize on the technologies available to them and enhance overall performance and effectiveness. However, exercise-based experiments are only possible with systems which already exist. The tenets of Exercise Red Flag explicitly exclude non-operational equipment, tactics, and programs barring approval by the air combat command overseeing the exercise [15]. Further, the tactics employed during these exercises are largely scripted and devoid of improvisation, potentially limiting the scope of explorations [76]. Taken together, these factors indicate the considerations for adversary tactics are confined to well-established boundaries, potentially incurring unwanted risk.

1.4 Summary

A review of the evolution of air combat over the past 80 years revealed a need in the system design process: To consider how tactical and technological factors might influence one another to effect outcomes. A search through the available literature showed the US DoD has acknowledged and grappled with this need for several years. However, the currents standards for tactical explorations appear to introduce undesirable amounts of bias or impose undue limitations on the design process by virtue of relying on human input. These observations substantiated the need for further research into how tactical considerations might be incorporated into the system design process.

Several challenges to the generation and analysis of alternative tactics were identified. Among these were the low amount of available knowledge in the early phases of the design

process, the risk of introducing unwanted bias, and the difficulty in capturing interactions between systems in operational scenarios. These observations lead to the formulation of the research objective.

Research Objective

The objective of this research was to enhance design space exploration for materiel solutions to capability gaps by enabling exploration of employment concepts to support the evaluation of value objectives in engagement-level analyses

1.4.1 Document Organization

This chapter established the research objective through observations made on the history of aircraft design processes. Chapter 2 explores the technical aspects of system design and behavior modeling in order to characterize the problem. Relevant theories from literature are synthesized, and gaps in existing techniques and methods are identified. These gaps lead to the statements of the primary research questions. Chapter 3 explores possible solutions to each gap through more targeted literature searches, again leveraging relevant theory and existing methods where possible. An attempt is made in Chapter 4 to combine the findings of Chapter 3 into a coherent methodology through alternative down selection informed by the literature and problem characterization. The result was a composition of hypotheses to each research question, each of which was tested in a sequence of experiments which built upon one another. Chapter 5 tests the overarching methodology as the primary hypothesis to satisfying the research objective. Finally, a discussion of findings and results is given in Chapter 6, along with some ideas for future work.

CHAPTER 2
PROBLEM CHARACTERIZATION

"We think that things we make can solve our problems, but our problems are much more complex than that."

— Malcolm Gladwell

The previous chapter established the need for and value of a capability to generate and evaluate alternative tactics in the early phases of the system design process. Key challenges were identified in the available literature. This chapter characterizes the problem of tactics explorations in the design process in greater detail, establishing necessary components of an approach to answering the motivating question.

2.1 Anatomy of an Analysis Methodology

The generic decision-making process shown in Figure 1.2 provides a useful, high-level perspective on design problems. However, it makes some concessions in specificity in order to be as generic as possible. The first three steps of the process are largely covered by CBA, where the needs are established, the problems defined, and the value established through the identification of capability gaps. Analysis of alternative tactics and how they interact with technology alternatives takes place in the next two steps: Generation and evaluation of feasible alternatives.

Generating alternative, potentially innovative tactics for evaluation is the focus of the motivating research question. However, the context of those evaluations must be established first. A more detailed architecture of the evaluation process was developed by Biltgen. His process, shown in Figure 2.1, provides some insights into the anatomy of a system design methodology [16].

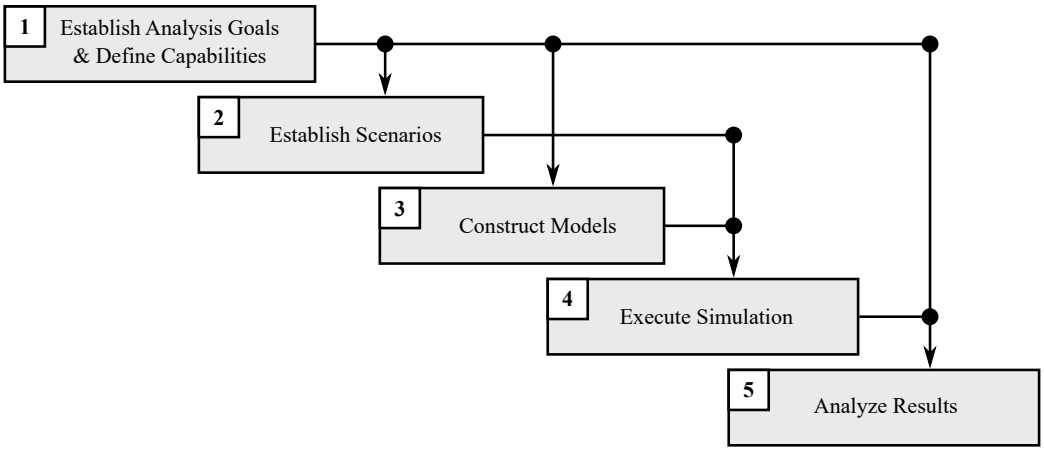


Figure 2.1: Biltgen's "common sense" quantitative technology evaluation process [16]

2.1.1 Design Space Exploration

The first step of Biltgen's process is to establish the analysis goals and define capabilities in terms of "the trade space of potential technology options" being considered [16]. This is known as design space exploration (DSE). Kang, Jackson, and Schulte define DSE as "the activity of discovering and evaluating design alternatives during system development" [68]. They identify three necessary elements to DSE: A representation of the design space, an analysis methodology, and a technique for exploration. These were precisely the elements required for behavior model exploration identified previously. These activities can be computationally intensive, particularly for sufficiently complex problems where evaluation is non-trivial.

Representing a Design Space

Design spaces can vary widely depending on the nature of the problem being considered. At the highest level is morphological analysis, which focuses on broad decisions about the design in question [163]. A Morphological Matrix can be produced, which identifies "the major functions or characteristics of a system on the vertical scale and all the possible alternatives (or system attributes) for satisfying the characteristics on the horizontal scale" [70]. A notional Morphological Matrix is shown in Figure 2.2 for a fighter aircraft design.

Alternative / Characteristic	1	2	3	4
Vehicle	Wing & Tail	Wing & Canard	Wing, Tail, & Canard	Wing
Minimum Combat Radius (km)	500	1000	1500	
Payload (kg)	6,000	8,000	10,000	
Maximum Mach	1.2	1.6	2.0	
Ceiling (km)	10	15	20	
Observability	Low	Standard		
Armament	Guns	Missiles	Guns & Missiles	
Weapon Bays	Internal	External	Internal & External	

Figure 2.2: Notional morphological matrix for a fighter aircraft

There are 5,832 possible combinations of characteristics which are possible using this simple matrix, although some combinations may be incompatible. For example, it would be impractical to place external weapon bays on an aircraft design for low-observability, since those protrusions would increase observability from certain angles [56]. Selection of a single combination of morphologies for the design process can leverage input from subject matter experts or feasibility analyses such as Technology Readiness Level or System Readiness Level [114].

Detailed Analysis of a Chosen Morphology

The next step after identifying an acceptable morphology for the design is to specify the *design attributes*, which are specifications of the concept in greater detail. Jet-powered aircraft attributes considered at this level could include wing planform area, thrust-to-weight ratio, thickness-to-chord ratios, and more [70]. Morphological alternatives represent discrete and categorically distinct alternatives. Design attributes can be continuous or discrete,

but are somewhat more narrowly scoped since they apply only to a single morphology at one time.

Capturing the effects of interactions between systems presents a non-trivial design challenge. The potential for complex interactions, emergent properties, and non-linear responses reduces the likelihood that an optimal design can be identified. Rather, the focus of design studies shifts towards assessing and analyzing the trade-offs between alternatives [69]. The goal of a DSE is to characterize the influences of different technologies or design variable settings on overall measures of effectiveness (MOEs) and measures of performance (MOPs). An MOE is a measure of "the impact of the actions of the [individual or system] and the [individual or system] on the effectiveness of achieving mission and task objectives" [91], while MOPs are used to determine if the system is "doing the right things to achieve the desired effect" [91].

Analysis on a given morphology can be conducted using a Design of Experiments (DOE) [70]. The purpose of a DOE is to identify a subset of all possible combinations for analysis which allow one to approximate and gain insights into possible trends across the space. That is, by sampling an appropriately distributed selection of points across the design space, the analyst can generate a representative model of the true response. DOEs are particularly useful when the true function is difficult or costly to evaluate, making exhaustive searches infeasible.

There are several types of DOEs used in literature. One of the simplest is the factorial design, which divides each dimension of the space into equally-spaced points and permutes over them. This can lead to extremely large designs of experiments, since the number of points is given by product of the discretization in each dimension. High-dimensional spaces or those which require fine discretizations quickly cause the total number of cases to explode, partially defeating the purpose of the DoE. Factorial designs offer certain guarantees, including uniform coverage of the space and sampling at the edges where interesting phenomena may occur. However, they also leave gaps in the space, especially for coarse

discretizations.

At the other end of the spectrum from the rigidly structured factorial design is the random sampling strategy. This approach generates points for evaluation using a random number generator. A uniform distribution is often used, but others are possible as well. A benefit to this method is that it samples interior points which a factorial design would not be capable to generating. However, there is a good chance that, with a low number of samples, there will be large gaps in the space which are not sampled. As with the factorial design, this can be overcome by generating more samples, but this comes at increased cost and so may not always be feasible.

The Latin Hypercube Sampling (LHS) strategy can be an effective technique for striking a balance between the factorial and random methods when sampling a high-dimensional space [65]. Similar to the factorial design, the LHS strategy discretizes each dimension into evenly-distributed points. However, rather than permuting over these sets, they are randomly sampled without replacement. This means that, for the same total number of samples, and LHS has finer resolution in any single dimension than a full factorial. Compared to a random sampling, it guarantees even sampling along each dimension.

Regardless of the chosen sampling technique, the sampled subset of design variable settings is carried through the rest of the analysis process shown in Figure 2.1. As the first step in the process, these design attributes will necessarily flow into any subsequent exploration of alternative employment concepts. This is necessary because tactics and technologies can have synergistic effects on MOEs. The Lockheed Martin F-35B is an excellent example of this: Technology which reduced the radar signature of the aircraft, coupled with enhanced communication capabilities, enabled pilots to employ what would have other been considered an extremely risky tactic to eliminate a threat much faster than expected [39]. This exposes a gap in the high-level process which must be addressed:

Gap 1

The potential effects of design attributes must be considered when exploring employment concepts

2.1.2 Modeling & Simulation

The "common sense" process casts the generation and evaluation of feasible alternatives as model construction, simulation execution, and analysis of results. The focus was on producing quantitative data to support decision-making processes. Necessary interactions and informational exchanges between elements of the model must be defined. Model refinement might be necessary if satisfactory agreement with expectations is not achieved. This can be the most time-consuming step in the entire process [16].

This template could be used to guide any design evaluation process. The third step was of particular interest because the model construction step would necessarily involve the identification of alternative tactics for analysis. However, the process is still too generic for the purpose of this work, leading to the second gap:

Gap 2

An appropriate modeling paradigm for exploring employment concepts is needed

Biltgen's process leverages on a specific class of evaluation techniques: Modeling & Simulation (M&S). Turner synthesized definitions for each of these two terms individually:

> **Definition:** A model is an abstraction of reality or one's concept that is used as an aid in answering a set of questions or to aid in communication [146]

> **Definition:** A simulation is the execution of a model [146]

There are many types of models which could be used for analysis. The most appropriate type of model or models employed for any given problem would likely depend on the questions being asked and the kinds of answers sought. The second gap had to be addressed early on because of the narrow scope of this effort. This formed the basis of the first research question: **(RQ1) What type of modeling should be used to facilitate exploration and analysis of alternative employment concepts?**

Distilling Selection Criteria

Selection criteria had to be established before the search for an answer to the first research question could begin. Five key criteria were identified. The model(s) must:

1. Be appropriate for early design evaluation
2. Provide quantitative performance information
3. Allow for exploration and innovation of tactics
4. Not introduce undue bias
5. Come at a reasonable cost to execute

Several types of models for experimentation were identified in Chapter 1. Exercise Red Flag is an example of a physical model constructed for the purpose of exploration and analysis of tactics. The exercise is an abstraction of reality in that the participants are not truly at war with one another. It also provides operators with an excellent opportunity to explore and innovate. However, these types of experiments fail to meet several other criteria. These experiments must be conducted with real systems, making them impractical for early design evaluation.

Physical experiments are limited in the amount of exploration they can facilitate. The time required to simulate an engagement in real time necessarily restricts the number of alternatives which can be evaluated. This can bias the experimentation process to only

consider tactics which appear promising from the start. This ties into the third failing: Physical experiments are expensive. Exercise Red Flag can cost up to US$20 million [149].

John Thach's match stick experiments are an example of conceptual modeling, also known as war gaming. These types of models do not rely on having the physical systems available, and experiments with them can be conducted very quickly. However, quantitative data can be difficult, if not impossible to elicit from these types of models because they reside solely in the minds of the modelers. Another consequence of this is that they may carry significant implicit biases which can be difficult to identify or eliminate. These factors make conceptual modeling ill-suited to the purposes of this work.

Computers can be used to construct models for analysis. Such models are frequently used at the engineering level of the hierarchy, such as for structural, aerodynamic, and thermodynamic analyses. Construct computer models can be a time-consuming endeavor, but they can enable rapid analysis once in place [16]. Computer models also allow for broad explorations by virtue of being virtual. The ability to simulate faster than real time allows more alternatives to be evaluated and reduces the barrier to conducting experiments outside of well-established borders.

Conjecture to RQ1

Computer modeling & simulation is the most appropriate paradigm for this work

Computer models are not without their drawbacks. Aside from the potential costs, constructing computer models must be done with care so as to ensure they provide adequate representations of reality. The limitations of physics are only imposed to the degree which the modeler chooses to implement them. The data produced by a computer model can only be trusted as much as the model itself. Model verification, validation, and accreditation exist to aid in establishing model credibility [151]. This matter will be addressed later.

Modeling Paradigms

Several distinct M&S techniques are available for answering a wide variety of questions. Three paradigms were found to be in common use in literature: system dynamics, discrete event, and agent-based. This led to the next, derived research question: **(RQ1.1) What type of computer modeling & simulation should be used?**

A thorough review of these M&S paradigms was presented by Borshchev and Filippov [22]. The review presented there is a synthesis of their findings with additional research on seminal works for each paradigm.

Discrete Event Simulation

The DES method was established by Gordon in 1961 [54]. It remains relevant to the modeling and simulation community nearly 60 years later because of its balance between simplicity and sophistication. The basic approach is to treat the evolution of the system of interest as a series of discrete events in time, hence the name. Each event corresponds to the occupation or liberation of resources by entities. This can be a very natural representation for many problems, including queuing and other processes.

The DES structure has several limitations when it comes to systems which evolve continually over time without clearly identifiable events in the intermediary. In the Thach weave, for example, the engagement begins and the engagement ends, and therefore the model may be viewed as only having two events. However, as the Thach weave itself shows, the intermediate states can be crucial to realizing outcomes. The DES structure might require those intermediate states to be abstracted in a way which does not adequately capture the desired effects. This enables DES to address large-scale problems but also hampers any explorations at more detailed levels.

System Dynamics

The SD approach was developed by Forrester in 1958 [45]. This approach casts the problem as a set of rates which are related to one another. That is, it describes the scenario as a coupled system of differential equations. Lanchester law equations (2.1) are an example of an SD model, where A and B are the strengths or sizes of opposing forces and β and α are rate coefficients determining the attrition of each side over the course of the engagement. Formulating the problem this way can enable the observation of secondary and tertiary effects, and beyond [44]. This is important because higher-order effects can be significant yet difficult to intuit and, therefore, predict. Importantly, SD is only useful for observing trends in the responses. It makes no claim as to the veracity of the values observed.

$$\frac{dA}{dt} = -\beta B, \quad \frac{dB}{dt} = -\alpha A \tag{2.1}$$

Some issues may be encountered with regard to mathematical tractability when using SD because it relies on solving differential equations. If the feedback loops lead to stiff systems then solving them can become very difficult or costly. Furthermore, SD is traditionally concerned with large-scale problems, e.g. the effects of corporate policy [46], and less on the details of how individual decision-making processes can effect outcomes. Implementations of this modeling approach may also face challenges when the systems of interest become extremely large and complex, leading to a high degree of connectivity and complex feedback structures.

Agent-Based Modeling

The agent- or individual-based approach has enjoyed growing popularity since the mid-2000s [79]. Agent-based models (ABMs) focus on interactions between entities and how those interactions effect change in the model. Macal and North present a list of criteria which would motivate the use of ABMs in their 2005 paper [78]. Among other criteria,

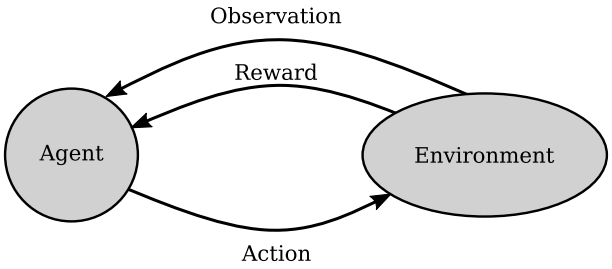


Figure 2.3: Agent-environment framework [73]

they argue that an ABM should be used when:

- There is a natural representation as agents
- Agents adapt and change their behaviors
- Agents learn and engage in dynamic strategic behaviors
- Agents have dynamic relationships with other agents
- Agents have a spatial component to their behaviors and interactions
- The past is no predictor of the future
- Scaling-up to arbitrary levels is important

The problem of identifying effective tactics can be naturally represented as an agent-environment interaction using the perspective developed by Legg and Hutter [73]. The framework of these interactions is shown in Figure 2.3. In this framework, the agent observes the environment and executes an action in response, after which the agent receives feedback from the environment in the form of a reward signal. The stimulus-principle-response framework of behaviors presented in Chapter 1 would reside entirely within the Agent element of this framework.

Agent-based modeling and simulation (ABMS) appeared to be preferable over DES or SD based on the criteria identified by Macal and North. All of the criteria listed above are satisfied by the problem of exploring tactics in engagement-scale operations. Explicit inclusion of behaviors, learning, and adaptation in the criteria strongly motivates the use of an ABM.

ABMs are commonly used in modern literature. A survey conducted by Allan in 2010 identified over 30 distinct software packages for implementing an ABM [5]. A 2017 survey by Abar et al. described and classified a number of ABM tools [1]. The availability of these methods across a variety of disciplines further supports the use of the agent-based approach. Furthermore, the combinations of Conjectures 1 and 1.1 serve to fill the second gap.

Conjecture to RQ1.1

Agent-based modeling & simulation should be used

2.2 Behaviors in Agent-Based Modeling

A wealth of literature has supported the use of agent-based models for exploration of and experimentation with alternative behaviors. However, the fundamentals of constructing agents and implementing behaviors are not consistent across disciplines. According to Bonabeau, the details of behavior model construction remain "an art more than a science" [21]. Bonabeau further explained that, when attempts were made to represent humans as agents in a computer model, there were significant hurdles to quantifying and capturing intangible factors. Such factors could include irrationality, subjectivity, and other measures of disposition.

Bonabeau's observations on the construction of behavior models for ABMs indicated the existence of a gap: There is no standard method for modeling organic decision-making processes. The ability to model decision-making in a defensible manner would be critical to conducting effective experiments in the context of this work. Proper descriptions of behavior models are necessary in order to realize a fundamental purpose of a model: To aid in communication. If the behaviors were not adequately described then effectively communicating any insights could become unnecessarily difficult. The results might be deemed unreliable or untrustworthy if the phenomenology were not deemed acceptable.

2.2.1 The ODD Protocol

Broad challenges to effective agent-based modeling were observed by a number of scientists in the field of ecology – the study of how organisms interact with one another and their environment. Grimm et al. proposed the ODD protocol to facilitate the description of behaviors and ABMs in general [57]. The protocol established a set of categorized descriptors which the authors argued should form the basis of any ABM implementation in any discipline. These descriptors were intended to provide a common means of documenting and substantiating the logic underpinning an ABM.

The first category, *Overview*, includes descriptors for: the purpose of the model; the variables, parameters, and scales of the model; and the overall process and scheduling at a high level. The second category, *Design Concepts*, included concepts of emergence, adaptations, interactions, and stochasticity. These concepts were covered more thoroughly in an earlier book by Grimm and Railsback [59]. The third category, *Details*, addresses the initialization of the simulation, the inputs to the model, and any submodels contained within the larger model.

ODD+D: Including Decision-Making Elements

Grimm et al. acknowledged that their protocol may have been incomplete and in need of updates as the scientific community adopted and implemented it [57]. To this end, they published an update in 2010 in which several elements of the protocol were revised or clarified. They noted some redundancy in the questions but argued the emergent benefits outweighed any potential costs.

They also acknowledge the need for community buy-in and feedback, and encouraged adaptation of the basic protocol to fit the needs of diverse sub-communities [58]. Müller et al. extended the ODD protocol to include *Decision-making* as a stand-alone category [87]. They motivated the inclusion of this category through critique of the baseline protocol:

1. Central aspects of human decision-making were not explicitly addressed
2. Theoretical and empirical bases for selecting decision submodels were not sufficiently emphasized
3. The *Design* concepts are not suitable for describing human decision-making

Müller et al. developed an additional set of protocol questions to be addressed by the modeler. These were aimed at eliciting information about the details of the decision-making models embedded within the ABM. The full list of questions to be answered by modelers adhering to the ODD+D protocol is extensive; there are 51 in all [87]. Many of the questions posed by the protocol were not directly applicable to this work. However, the protocol did provide a useful perspective on the behavior model construction process.

A conceptual model of organic behavior was presented in Chapter 1. Distillation and synthesis of various definitions from literature was used to establish the stimulus-principle-response architecture shown in Figure 1.4. The architecture must be translated into the language of computer ABMs to be useful in meeting the stated research objective.

One question from the ODD+D protocol indicates this need directly: *(II.ii.c) How do agents make their decisions?* This question "refers to the way in which the rationality behind decision-making is translated into the specific decision-making rules" [87]. That is, this questions asks the modeler to identify the principles employed by the model – the rules or beliefs governing the behaviors. Furthermore, the protocol asks: *(II.i.b) On what assumptions is/are the agents' decision model(s) based?* and *(II.i.c) Why is/are certain decision model(s) chosen?* These questions can be addressed if a sufficient theoretical basis for the chosen models can be established. This exposes the third gap to be addressed:

Gap 3

A theoretical foundation for exploring and analyzing employment concepts is needed

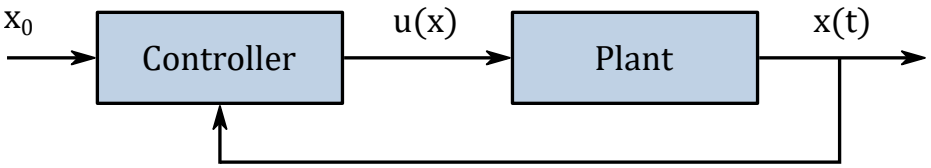


Figure 2.4: Closed-loop control system diagram

2.2.2 Establishing a Theoretical Basis

The search for relevant theories to support the creation and experimentation processes for behavior modeling in an ABM began with the purpose behavior model serves. At a fundamental level, agent behaviors can be understood as controllers. This becomes readily apparent when comparing the agent-environment interaction diagram shown in Figure 2.3 to the block diagram of a closed-loop control system shown in Figure 2.4. The agent is analogous to the controller, and the environment to the plant. Further, the function $u(x)$ can be viewed as an abstraction of the stimulus-principle-response conceptualization of behaviors shown in Figure 1.5. The two may also be compared by their definitions. Brogan defines a control system as one which "exists for the purpose of regulating or controlling the flow of [resources]" [25]. This is in line with the definitions of behaviors given in Chapter 1.

Viewing agent behaviors as control systems allows the former to be described using the language of control theory. A review of the available literature provided insights into the process of constructing an optimal controller.

Formulations of the Optimal Control Problem

Brogan proposed a formulation of the optimal control problem from a high-level perspective. He identified four components of the design process for an optimal controller [25]:

1. A description of the system to be controlled
2. A description of system constraints and possible alternatives
3. A description of the task to be accomplished

4. A statement of the criterion for judging optimal performance

Optimal control theory traditionally describes the system to be controlled as a set of equations defining how the system evolves over time as a function of its state and any control inputs applied to it, for example using the continuous or discrete time formulations (2.2) [25]. Defining the system to be controlled establishes the states of the system which may be observed and used by the controller to achieve its goal.

$$\dot{\mathbf{x}} = \mathbf{f}(\mathbf{x}(t), \mathbf{u}(t)) \quad \text{or} \quad \mathbf{x}(k+1) = \mathbf{f}(\mathbf{x}(k), \mathbf{u}(k)) \tag{2.2}$$

Constraints may exist on the description of the system, such as limits on the possible values of states or controls. These may come from first principles, physics, or other sources.

A description of the task to be accomplished can take many forms. Brogan provided two examples. The first was transitioning from an initial state $\mathbf{x}(t_0)$ to a desired final state $\mathbf{x}(t_f) = \mathbf{x}_d$, where the end time t_f may be a specific value or the minimum possible [25]. Brogan's second example was transitioning from any possible initial state to a specified region within the state space.

Brogan identifies a general-purpose form of performance criterion, the discrete form of which is given by (2.3) [25]. The real, scalar-valued functions S and L indicate the cost associated with the terminal and transient states, respectively. The goal of the controller is to minimize the sum J, and alternative controller designs can be compared using this quantitative metric.

$$J = S(\mathbf{x}(N)) + \sum_{k=0}^{N-1} L(\mathbf{x}(k), \mathbf{u}(k)) \tag{2.3}$$

Locatelli built upon the four basic elements of controller design to establish a more-detailed process. He identified six components to the structure of an optimal control problem [75]:

(a) The equations which describe the dynamic behavior of the system

(b) The set of allowable initial states

(c) The set of allowable final states

(d) The performance index

(e) A set of constraints on the state and/or control variables

(f) The control interval

This new structure adds the identification of allowable final states and a control interval. The set of allowable final states "can entail the complete or partial specification of the final state as well as its freedom" [75]. This is an important component of optimal control theory because analysis of the costate equations requires knowledge or specification of the final state. However, the exact final state may be unknown, hence the caveats regarding partial specification and freedom of the state to vary within the feasible space.

The control interval is the window of time within which the controller may interact with the system. This window may be specified a priori or determined by the controller. Furthermore, the window may be finite or infinite in duration. This could have been interpreted as a kind of constraint on the controller, but differed slightly from the state and control variable constraints and so warranted its own place in the process.

Limitations of Optimal Control Theory

The theory of optimal control can be a powerful tool for solving complex problems. The Hamilton-Jacobi theory and variational methods establish sufficient and necessary conditions for optimality, respectively [75]. These mathematical methods can enable designers to construct provably optimal controllers for their problems.

Solving the requisite systems of equations often involves a significant amount of mathematical manipulation, and sometimes requires strong assumptions in order to be tractable. For example, Locatelli's description of the optimal controller design problem includes the identification of admissible final states. This would require the designer to know or oth-

erwise constrain the set of all final states a priori. This may not be possible in all circumstances, reducing the utility of these methods.

The process of designing an optimal controller may be further complicated by the complexity of the system of interest. Design of an optimal control becomes increasingly difficult as the dynamical behavior of the system becomes increasingly non-linear or erratic. Solution of a two-point boundary value problem often cannot be done analytically in these cases, and expensive iterative numerical techniques may be required.

Kirk identified three iterative methods for finding optimal controllers when analytic solution is not feasible: Steepest descent, variation of extremals, and quasilinearization [71]. Each of these methods presents at least one significant challenge to analysis of complex systems. The method of steepest descent requires an initial guess for the control history, and only guarantees a local optimum. Variation of extremals requires a guess for the initial value of the costate for integration, and the results of this method can be sensitive to that guess. Quasilinearization involves approximating the non-linear system of questions using a linear one, and iteratively improving that approximation. However, it may be difficult to develop a linear approximation of the system, especially if there are discontinuities.

Synthesizing a General Process

Despite its limitation, optimal control theory provides a solid theoretical foundation for behavior model construction in compliance with the ODD+D standard. A general process for the model construction process could be distilled from the those established by optimal control theory. Furthermore, each step in the process should map to specific questions from the ODD+D protocol.

The first step from both Brogan's and Locatelli's processes were the same and, therefore, could be consolidated. The first step in the synthesized process would be: **(1) Describe the system of interest**. In the context of an ABM, describing the system would involve identifying the agents in the model and interactions between them, as well as how

they and their interactions effect change in the environment.

The next step was identified by examining Locatelli's process more closely. His problem structure entails the identification of allowable initial and final states, as well as constraints on the intermediate states. These three components could be collapsed into one step: **(2) Identify the observable state space**. An agent's observable state space encapsulates all possible permutations of stimuli which that agent might have to respond to.

The next logical step, after having described the system and the observable states, would be to establish how the controller might influence its state. The next step in the synthesized process was then: **(3) Identify the admissible controls**.

Both Brogan and Locatelli identify the need for a performance index to facilitate quantitative comparison of alternative controllers. This establishes the next step in the process: **(4) State the performance index**. The performance criterion (2.3) would be a good candidate for this step in most cases.

Both processes found in literature relied on analytical techniques to solve for optimal controllers. However, it was shown that such solutions might not exist or may not be easily found in all cases. Kirk showed that, for such cases, there is some experimentation which must be performed on the mappings between observable states and admissible controls in order to identify a locally optimal model. These observations necessitated the inclusion of a new step in the process: **(5) Experiment with controllers**.

The last step follows naturally from the preceding one. The results of the experimentation effort should be leveraged to make an informed decision. The final step in the generic process is: **(6) Select a best controller**.

The generic controller construction process can be viewed as an application of the generic decision-making process to the specific problem of creating and selecting from a set of feasible alternative controller designs. Several steps in the process could be populated with methods from literature. The ODD+D protocol and the works by Brogan, Locatelli, and Kirk offer several approaches to the first four steps.

As with the generic decision-making process, generating and evaluating feasible alternatives present significant hurdles which must be overcome in order for the process to be effective. The risks identified by Alberts and Hayes apply to these steps, just as they did before. However, the scope of the problems surrounding these processes has been reduced through the identification of ABMs as the most appropriate modeling paradigm and the controller design process as the theoretical basis for model construction. This reduced scope allows for more targeted investigations into how those experiments might be conducted.

A Framework for Behavioral Learning

Skinner proposed a theory of how organisms, including humans, learn behaviors, a process he termed *operant conditioning* [129]. The process involves the organism establishing associations between stimuli and responses which are either reinforced or extinguished based on feedback received from the environment. If the feedback is rewarding or otherwise desirable then that responses to said stimulus becomes more likely; if the feedback is punishing or undesirable then the response becomes less likely. The complexities of brains, human or otherwise, make any attempts at explaining *how* the reinforcing and extinguishing mechanisms occur, but Skinner was not concerned with such technical details.

Operant conditioning may also be referred to as *learning*. Müller et al. defined learning as the process of making "changes in ... one's decision-making rules" [87]. In the context of behavior modeling in a computer ABM, learning would mean making changes to the function or functions which make up the behavior model, ideally so as to realize better performance.

A generic framework can be established as a template for the experimental process for creating behavior models based on the concepts of operant conditioning and learning in ABMs. Skinner distinguished two types of conditioning: Type S and Type R [130]. Conditioning of Type S involves the organism developing associations between paired stimuli, occurring one after another, and eliciting a response to the first in expectation of the sec-

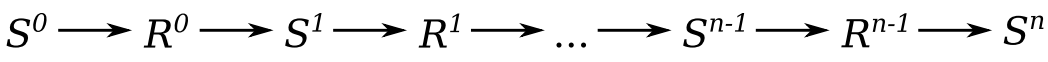


Figure 2.5: Pairs of stimuli and responses used for conditioning behaviors

ond. Conditioning of Type R, on the other hand, occurs when an organism associates the elicitation of a response with the realization of desirable feedback. Type R is more relevant to the present research effort than Type S, since the latter is a passive process compared to the active approach taken by the former.

Conditioning of Type R takes place on sequences of paired stimuli and responses, as shown in Figure 2.5. The organism is then given rewards or punishments, and the elicitation of responses is modified so as to realize a sequence of stimuli and responses which are expected to result in higher rewards or, equivalently, the lower punishments.

A sequence of stimuli and associated responses by an organism is equivalent to a simulation of a behavior model in the context of computer ABMs. Conditioning of Type R is the process by which feedback obtained through that simulation is used to modify the behaviors. This allows for the formulation of a generic framework for learning, shown in Figure 2.6. How the models are initialized will likely depend on what information is available to the modeler. If expert knowledge exists about how an agent might behave then that knowledge may be used to construct an "educated guess". However, if no expert knowledge is available then a random initialization may be warranted. Random initialization might also reduce the risk of unnecessarily confining explorations to well-established borders, albeit with the potential for requiring some trial-and-error.

Simulating the model is straightforward: Stimuli are collected, the agent is tasked with making a decision about its response, and the model advances in time. This process repeats as necessary until termination criteria are met. The update step is of primary concern to the present discussion. This would be where conditioning occurs and the rules of the behavior model modified based on the information collected through simulation.

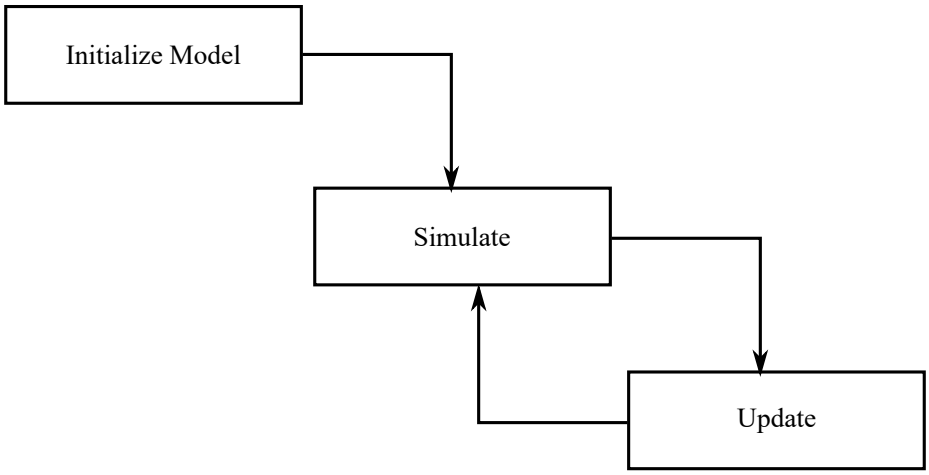


Figure 2.6: Generic framework for learning by conditioning

2.2.3 Challenges in Controller Experimentation

There are two well-known challenges to experimenting with controllers: The curse of dimensionality and the temporal credit assignment problem [112, 137]. These challenges can compound one another in the process of exploring a space of behaviors.

The curse of dimensionality can be understood through a simple example. Suppose there are three possible responses a system can elicit to any given stimulus. The trajectory of stimuli and responses shown in Figure 2.5 would have 3^n possible permutations or paths the agent could take through the space; if $n = 10$ then there would be $3^{10} = 59,049$ alternatives to analyze. Multiple trajectories would have to be samples to gain any insights into how performance might be affected by responding to any given stimulus in different ways. Exhaustive analysis would likely be infeasible because the number of possible trajectories increases exponentially with the number of steps n. This is the curse of dimensionality [112]. It would be ideal if a representative subset of all trajectories could be sampled for analysis. However, it may be difficult to select such a subset for analysis and comparison if different paths are sufficiently distinct in their outcomes.

The temporal credit assignment problem can simultaneously be a cause and consequence of the curse of dimensionality. The problem is thus: If the reward r^t is received at

time t, then how can credit (or blame) be assigned to all responses $R^{\tau}, \tau \in [0, t-1]$ [137]? That is, how can the organism decide which responses merit modification if time delays are present in the reward mechanism? If the organism were only concerned with maximizing its expected reward immediately after a response then this would not be an issue; the curse of dimensionality would be of little to no concern because the organism would have no incentive to explore the space more thoroughly. In the previous example, only 3×10 stimulus-response pairs would have to be analyzed since all responses could be tested to identify the best at each state.

The curse of dimensionality and temporal credit assignment problem can combine to make behavior exploration extremely difficult. The ideal experimentation process would be able to mitigate the challenges caused by both by facilitating broad explorations of alternative paths through the space of alternative decisions and by accounting for the temporal relationships between decisions at different states.

2.2.4 Experimenting with Models of Behavior

A review of available literature established the process of exploring employments concepts as, fundamentally, one of experimentation. There are two key components to any effort of experimentation: An apparatus and a process. The experimental apparatus is the object of experimentation – the thing on which experiments are being conducted – while the process describes the nature of the experiments, how they are carried out, and what information is sought from them. In the context of this research, the apparatus is the model or models of behavior within an ABM, specifically how agents make decisions about their actions based on their observations of the environment. These decision-making processes can take many forms in a computer model, and the chosen form must be amenable to some form of manipulation which facilitates the experimentation process. These observations exposed two further gaps, derived from the third:

Gap 3.1

A technique to allow agents to map observed states to admissible actions is needed

Gap 3.2

A process for exploring and evaluating different state-action mappings is needed

2.3 Experimentation at the Engagement Level

Engagement-level analyses are largely characterized by the interactions between entities or systems within the model. Changes to those interactions can effect different outcomes and, as a consequence, alter the course of an analysis effort. This may hinder discovery experiments and other efforts to explore employment concepts. Altering the employment of any one system in the environment may have unintended and unpredictable effects on the performance of other, related systems.

The variety of section tactics for two-on-one engagements described in Shaw's *Fighter Combat* are good examples of how changes in employment concepts can dramatically alter the sequence of events. This is most notable in the two versions of the half-split maneuver [123]. The setup for both is identical: An aggressor approaches a pair of fighters flying abreast from behind. Once the aggressor is noticed, the left fighter turns hard to their left while the right fighter continues straight ahead. What happens next depends on which fighter – turning or extending – the aggressor chooses to pursue. If the turning fighter is pursued then the extending fighter turns hard towards their wingman. The turning fighter, meanwhile, tightens their turn in an attempt to thwart the attack and drag the aggressor into the line-of-sight of the extending fighter. If, however, the aggressor pursues the extending

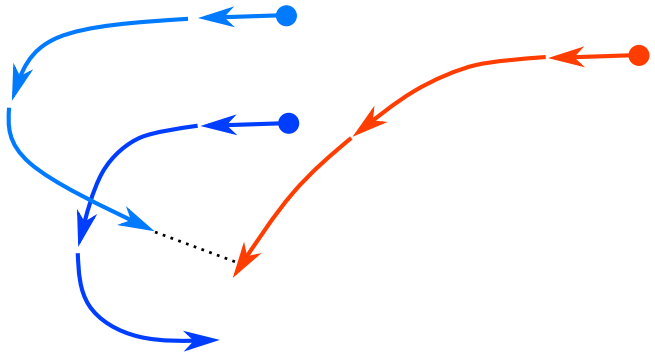
fighter then the latter continues to extend and turn slightly, dragging the aggressor into the line-of-sight of the turning fighter who has reversed their turn. These two scenarios are shown in Figure 2.7.

The two variations on the half-split defensive maneuver showcase the dynamic nature of engagement-level analyses. Agents have to consider the possible actions of other agents, and they may have to dramatically alter their course of action in response to a changing environment. Consider the behavior of the turning fighter, represented by the lower, dark blue lines in Figure 2.7. There could be two distinct categories of behaviors to explore, one for each choice of which fighter the aggressor chose to pursue. This would not present significant additional challenges if the behaviors of the extending fighter and aggressor were assumed to be known and unchanging in each case. However, such assumptions would be difficult to justify; Why should one agent be allowed to explore different behaviors and others not? Furthermore, the choice of behavior employed by the other agents could be called into question under such strong assumptions. Ideally, all agents would have their own dynamic behaviors as the environment unfolds over time. This exposes the last gap:

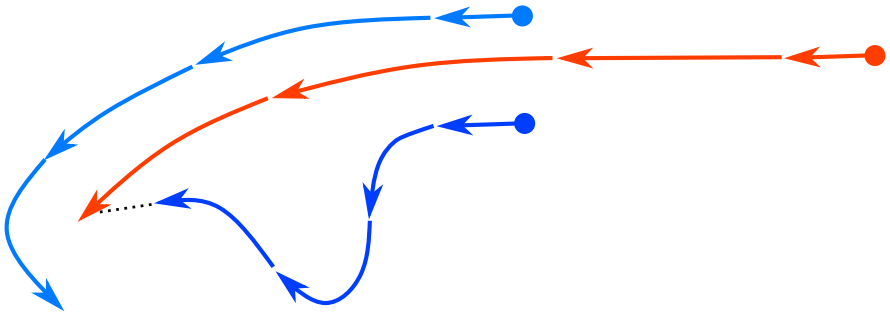
Gap 4

A technique for facilitating exploration and evaluation of employment concepts for multiple interacting agents is needed

There two main caveats to the fourth gap: Relevance and feasibility. Some agents might not be relevant to the employment concept exploration process, such as those whose behavior has already been explored or is well-understood. Alternatively, there might exist agents whose behavior cannot be subjected to these types of explorations, such as those whose behavior is governed by deterministic processes. Message routing or fire control might fall under these umbrellas. This relates to the second caveat, since defaulting the behaviors of certain agents will reduce the dimensionality of the behavior space to be explored. This can help to mitigate uncertainties in the model and focus efforts on those agents – hopefully few in number – whose behaviors might not be established or well-understood. Otherwise,



(a) Aggressor engages turning fighter



(b) Aggressor engages extending fighter

Figure 2.7: Two versions of the half-split maneuver, showing how the engagement plays out based on which of the two fighters (light and dark blue) the aggressor (orange) chooses to engage. Adapted from [123].

excessive costs might be incurred over the course of the experimentation effort. The confinement of this research to the scope of engagement-level analyses lessens the potential for feasibility constraints to adversely impact the effort, at least to some extent.

2.4 Review of Existing Methodologies

Several methodologies for quantitative analysis of capabilities have been developed and employed over the past two decades. Summaries of five works, selected based on their relevance to the research objective and demonstrated capabilities, are presented in the following sections.

2.4.1 Automated Combat Maneuvering

Austin et al. developed a maneuvering logic system for simulation of air-to-air combat engagements which they showed could be used for technology evaluation [8]. The methodology consisted of decomposing the engagement into a sequence of decisions made by each agent in the model. Those agents would conduct small simulations for each combination of decisions they could make in order to calculate a preference metric over the space of possible actions. They would then select their most-preferable action using a minmax algorithm. Variations in the update intervals and forecast times could be used to influence the decision-making processes and their effects on the simulation.

The use of small-scale simulation in the decision-making models allowed the technique to be applied broadly, since the effects of technology settings would be captured by those simulations and, therefore, would influence the action taken. However, this technique would not scale well since the matrix of action combinations must be evaluated exhaustively. This was not an issue in the one-on-one scenario considered by Austin et al., which had 49 cells, but the size of the matrix would grow exponentially as more agents were added, quickly making exhaustive evaluation infeasible.

2.4.2 Capability-Based Technology Evaluation for Systems-of-Systems

The first methodology identified was the Simulated-based, Object-oriented, Capability-focused, Real-time Analytical Technology Evaluation for Systems-of-systems (SOCRATES) methodology developed by Biltgen [16]. The methodology consists of 10 steps, derived from the five "common sense" steps. The first three steps involve setting up the analysis process, establishing relevant scenarios, and identification of appropriate M&S tools to support evaluation.

Step four maps high-level strategic objectives to lower-level operations using Quality Function Deployment. This allows some degree of tactical exploration by altering the relative importance and, therefore, priority of different operations. In Biltgen's example of an offensive air-to-ground campaign, an analyst could compare the tactical and strategic effects of eliminating enemy SAM or radar sites before conducting strike sorties.

Step five sees the creation of a "meta-general" computer model with the purpose of mitigating the need for human involvement in the analysis process. The model performs the functions of assigning tasks to assets based on capabilities – i.e. technologies – with the intent of maximizing expected performance and effectiveness with respect to current strategic objectives. This is accomplished using a surrogate model trained on simplified scenarios with sampled combinations of assets, tasks, and technology settings. The model can also be trained to consider elements of the environment, such as spatial or temporal factors. Biltgen found that artificial neural networks outperformed other options for creating surrogate models to serve as a meta-general.

The meta-general assigns tasks to its subordinates based on expected performance and effectiveness. The sixth step in SOCRATES is to create the behavior models of the subordinates which they employ when executing their assigned tasks. He noted that "behaviors are difficult to quantify for multi-agent systems where the evolving behaviors of individual agents confound the actions of each other" [16]. This further substantiates the earlier observations made on multi-agent systems and the challenges they pose to the behavior

modeling process. Biltgen's process overcame this difficult by employing a "playbook" approach, where discrete employment concepts are defined and evaluated by agents within the simulation. A surrogate model of expected performance for each employment concept as a function of the assigned task and design variable settings can be constructed. Evaluating the function allows agents to make quasi-intelligent decisions based on their current capabilities.

Biltgen's methodology enables agents at multiple levels to exhibit different behaviors in response to changing environments and design variable settings. However, it does not allow for broad explorations within the space of tactical alternatives because the employment concepts for each agent are prescribed by a playbook. He acknowledged the potential for artificial intelligence and machine learning techniques for provide the necessary capabilities for developing "new maneuvers or tactics based on real-time learning and adaptation" [16] but cited reasonable concerns about the immaturity of the field at the time in choosing a different path. However, significant progress has been made in the area since the formulation of SOCRATES, making such approaches more feasible now than ever before.

2.4.3 Quantification of Doctrine

Tangen developed a method for quantifying doctrine to facilitate analysis [141]. Quantification is achieved through a functional decomposition of the mission or missions being analyzed. This allows for identification of alternatives for satisfying those functions, which can then be converted into quantitative values by enumeration. More points might have to be sampled in order to cover the increased dimensionality.

Tangen's methodology is similar to Biltgen's in that the tactical alternatives are prescribed. However, there are some significant issues with the quantification approach. Tangen provided an example in the form of a patrol mission over an area of operations. Assets surveil the area by flying routes over it. Possible routes were random walk, parallel search, orbit, and border patrol patterns. These are categorically distinct tactics for conducting

surveillance, meaning the corresponding doctrine parameter is nominal, rather than ordinal. That is, one cannot leverage the evaluation of the doctrines corresponding to index 1 and 3 to determine an expected value for that at index 2; there is no such relation between the value of the doctrine parameter and the corresponding tactics. An analyst may be forced to evaluate the space of doctrine parameters exhaustively in order to develop an acceptable model of performance as a function thereof.

Tangen's methodology relies on functional decomposition to identify potentially novel tactics. He argued that decomposing doctrine into lower-level concepts, similar to the concept of morphological decomposition, can allow for synthesis of new employment concepts. While this would allow for innovation and exploration, it necessarily constrains the space of alternatives which can be considered. Furthermore, creating extensive lists of alternative functions will adversely impact the analysis effort by imposing higher burdens on computational resources.

Quantification of doctrine can aid in answering the question, *"Given a set of technologies and tactics (i.e., doctrine), what combination has the highest expected performance and effectiveness?"* However, it does not directly facilitate answering the question, *"Given a set of technologies, what are the tactics which maximize expected performance and effectiveness?"* The latter makes fewer assumptions about the nature of possible tactics and what might constitute a highly effective employment concept.

This method may allow for explorations of tactics if the doctrine parameters of all relevant systems can be quantified. However, the issues of a nominal doctrine parameter spaces, increased computational burden, and maximizing expected values would be compounded by extending the method to include more agents. In Tangen's surveillance example, one might want to explore how search strategies perform against an opponent using different strategies to avoid detection. One would then have to identify, quantify, and implement the various avoidance strategies, likely using the same approach as was employed for the search strategies. The cost to evaluate these alternatives would increase as the ex-

ploration of doctrine parameters increases in a combinatorial manner. All of these factors together may make quantification of doctrine ineffectual for exploring the combined space of tactics and technologies.

2.4.4 The Stochastic Agent Approach

Gordon developed a method for exploring employment concepts based on Monte Carlo simulation and high-throughput computing [55]. His Stochastic Agent Approach was motivated by observations similar to those made on Biltgen's and Tangen's methodologies, namely that the process of defining doctrine at a level appropriate for infusion into an ABM quickly became prohibitively intensive and expensive as models grew more complex. His solution was to represent agent decision-making using random processes and leverage massive, embarrassingly parallel computing capabilities to characterize the distribution of MOEs and MOPs over possible combinations of decisions.

Gordon's method resolves several issues seen in Biltgen's and Tangen's. Greater explorations into the space of possible tactics are made possible by the implementation at the level of agent decision-making. This may allow for more dynamic behaviors in the simulation than would be possible with the prescriptive approaches utilized by earlier works. Gordon showed his method "not only enabled more effective exploration of the Mission Space, but also could generate mission plans similar to those from more costly optimization approaches" [55].

Like Tangen's methodology, Gordon's method can be implemented on all agents in an ABM. This would aid in exploring the combined space of employment concepts by allowing potentially adversary agents to exhibit different behaviors. However, those behaviors would not capable of accounting for any expectation of possible actions by other agents and, therefore, unable to respond effectively or meaningfully. This may be a serious flaw in the methodology.

A major drawback to SAA is the reliance on Monte Carlo simulation, which is essen-

tially a trial-and-error method. The agents cannot be said to possess any form of rationale, and make their decisions solely based on arbitrary probability distributions. The implementation of decision-making parameters in SAA is also orthogonal to the design space parameters, meaning there is no direct mechanism by which design attributes can be factor into the decision-making. This may facilitate exploration, since it allows the decisions to be independent from design attributes, but prevents exploitation of effective behaviors. There is no semblance of learning; effective strategies are necessarily identified purely by chance. This constitutes a significant risk to the analysis process, per Alberts and Hayes.

2.4.5 Mission-Level Weapon System Analysis

Connors created a methodology for analyzing weapon system alternatives in mission-level analyses which incorporated engagement-level simulations [29]. Behaviors were enabled through the use of behavior trees, which allowed agents to react to an evolving environment by dynamically selecting from a set of prescribed, established tactics for a two-ship section. This architecture allowed agents to seamlessly switch between offensive and defensive behaviors based on the detection of incoming threats.

An application of Connors' methodology to the analysis of combinations of missile technologies demonstrated the importance of capturing interactions between tactics and technologies. Their analyses showed tactics could alter the statistical measures of effectiveness at the mission level, and that preferential selection of certain tactics could adversely impact system metrics. They also showed how the implementation of dynamic behaviors augmented capabilities by allowing agents to capitalize on an ability to engage at greater ranges or from positions which would be more difficult to counter.

A significant drawback to Connors' methodology was the prescription of tactical alternatives in the model construction phase. The use of behaviors trees allowed for some flexibility but relied on having predefined maneuvers to select from, proper implementations thereof, and the overall structure of the decision-making model. The choice of finite

behaviors to include necessarily introduces bias, while the structure of the behavior tree and precondition values used might restrict the variability of behaviors which can be exhibited by agents in the model. Lastly, Connors' methodology only considered one side of the dynamic engagement; static behavior models were used on the systems opposing those to which the technologies were being applied.

2.5 Summary

This objective of this research was to enable the exploration of employment concepts to augment the system design process. Motivation came from observations on historical and contemporary accounts of tactical innovation by human pilots in air combat engagements. Furthermore, it was observed that failing to adequately consider the range of possible employments of new and novel technologies can adversely impact performance.

Four main gaps and two derivative gaps were found in the course of characterizing the problem of interest through a literature review. Gaps were exposed through observations on the generic nature of the common-sense process for quantitative technology evaluation and the need to populate the steps involved. The gaps are restated below for convenience. Gap 2 is included here for completeness, even though it has been addressed by findings from literature.

Gap 1: The potential effects of design attributes must be considered when exploring employment concepts

Gap 2: An appropriate modeling paradigm for exploring employment concepts is needed

Gap 3: A theoretical foundation for exploring and analyzing employment concepts is needed

Gap 3.1: A technique to allow agents to map observed states to admissible actions is needed is needed

Gap 3.2: A process for exploring and evaluating different state-action mappings is needed

Gap 4: A technique for facilitating exploration and evaluation of employment concepts for multiple interacting agents is needed

Three methodologies were identified which sought to incorporate considerations for alternative employment concepts into the technology evaluation process. The developments of these methodology contributed to the body of knowledge in this area. Biltgen's SOCRATES showed how the decision-making models embedded in agents within an ABM can be parameterized by the design variable settings to produce more flexible models. Tangen's approach proposed the use of functional decomposition to explore alternative employment concepts. Gordon's SAA extended the functional decomposition of doctrine to the level of individual decision-making and facilitates explorations at that level.

Each of the methods failed to address at least two of the identified gaps. Biltgen's did not adequately address Gap 3 because the agent behaviors did not have solid theoretical bases, instead relying on subject matter expertise and hand-crafted functions to enable quasi-intelligent decision-making. This would make attempts at manipulating those behaviors more difficult, and impose significant challenges in attempting to apply the methodology simultaneously across multiple interacting agents. These constitute shortcomings with respect to Gaps 3.2 and 4.

Figure 2.8: Comparison of existing methodologies with respect to identified gaps. Green indicates the gap would be adequately addressed, yellow indicates partial fulfillment, and red indicates the gap is not addressed.

Tangen's methodology would allow for explorations of how alternative employment concepts and design attributes could produce synergistic effects, but does not allow the design attributes to directly influence employment concepts. This leaves Gap 1 at least partly unaddressed. Functional decomposition provides some theoretical basis for the choice of employment concepts implemented, but still relies on human input and does not fully satisfy Gap 3. Furthermore, the types of employment concepts utilized by Tangen's methodology do not make room for dynamic interaction and manipulations of the principles underlying those interactions. Lastly, applying a functional decomposition to every agent in a model and exploring the resulting DOE would likely be very costly. Based on this, it was deduced that Tangen's methodology would not adequately address Gaps 3.1, 3.2, and 4.

Gordon's SAA has similar shortcomings as Tangen's, in that design attributes do not directly influence the decision-making processes employed by agents. Similarly, agent decisions are not produced by controllable models which can be easily manipulated to discover new employment concepts. However, the stochastic decision-making processes would be relatively easy to implement across multiple agents. A summary of how the existing methodologies perform with respect to the identified gaps is shown in Figure 2.8.

2.5.1 Outline of the New Methodology

A template for creating a new methodology was synthesized from the literature reviewed in this chapter. The template began with the generic quantitative technology evaluation pro-

cess created by Biltgen, shown in Figure 2.1, which is itself a reformulation of the generic decision-making process shown in Figure 1.2. The focus was turned to the third step, model construction. Agent-based modeling was identified as the most appropriate computer M&S paradigm to meet the research objective, and targeted literature showed optimal control theory could provide a theoretical basis for constructing models of behavior. The generic process described in Section 2.2.2 was synthesized from available literature. Experimentation with controllers was identified as the least-established step in the process for the types of problems being considered by this research. A simple framework for behavioral learning, shown in Figure 2.6, was adopted from the operant conditioning theory of behavior to address this gap. The synthesis of these various processes into a single, overarching methodology is outlined in Figure 2.9, where the gaps found in the literature are identified.

2.5.2 Statement of Research Questions

Multiple gaps identified in the problem characterization could not be adequately addressed by existing methodologies. This established a need for a new methodology to be created; one which can address the identified gaps. Four research questions were derived from the identified gaps. The first, derived from Gap 3.1, was: **(RQ2) How should state observations be mapped to actions in an agent-based model to facilitate experimentation with employment concepts?** The next research question followed directly from this and Gap 3.2: **(RQ3) What process should be used to manipulate the parameters of the state-action mappings?**

Research questions 2 and 3 focus on low-level aspects of the behavior modeling problem. The fourth research question adopts the broader perspective necessitated by Gap 4: **(RQ4) How should explorations of employment concepts be conducted at the engagement level of analysis?** The last research question was aimed at Gap 1 and the overarching research objective: **(RQ5) How should explorations of employment concepts account for variations in design attributes?**

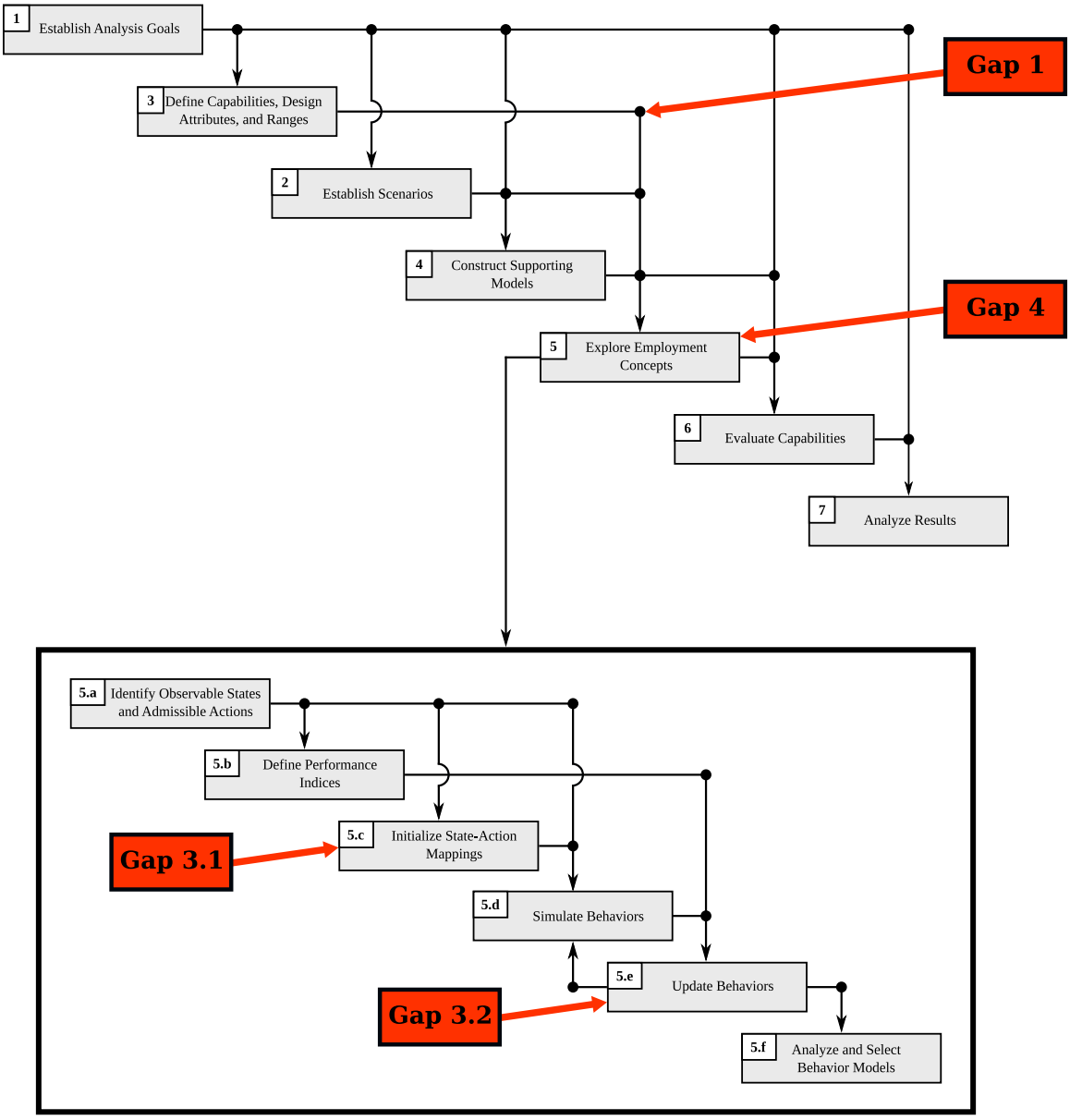


Figure 2.9: Outline of the proposed methodology

The relationships between the topics covered in this section, the gaps identified, and the stated research questions are shown in Figure 2.10.

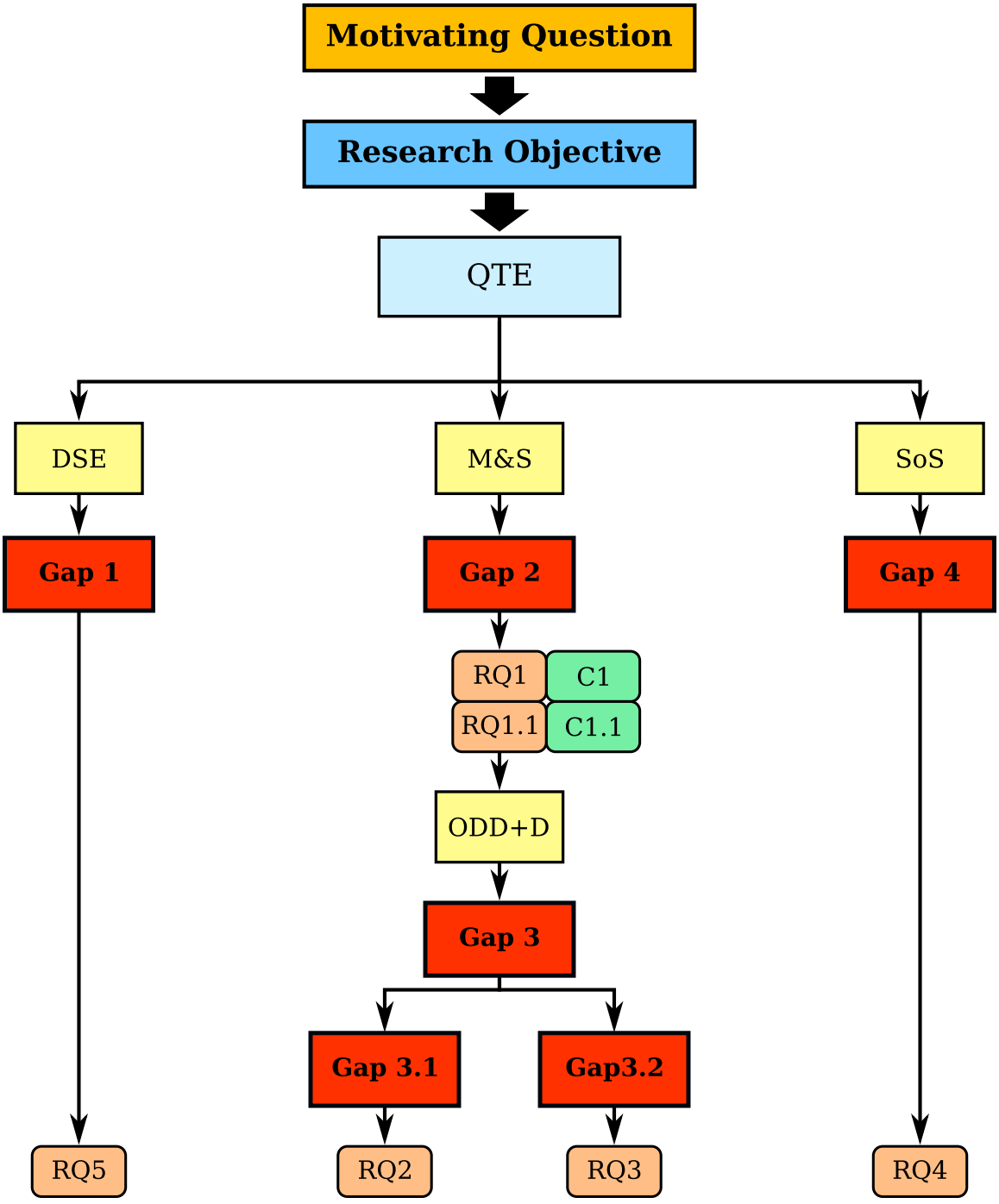


Figure 2.10: Overview of identified gaps and associated research questions

CHAPTER 3
ADDRESSING GAPS

"If something is presented as an accepted truth, alternative ways of thinking do not even come up for consideration."

— Ellen Langer

The second chapter established the need for a new methodology to enable exploration of alternative employment concepts in the system design process through a broad literature search. This chapter will present findings from detailed literature searches pertaining to each gap and associated research question. Candidates for answering the research questions and, by extension, addressing the gaps will be identified and examined.

3.1 Gap 3.1: Mapping States to Actions

Two candidates for state-action mapping came from the literature reviewed so far. The first was algebraic functions, which are commonly used in applications of optimal control theory. The second was decision trees, which were mentioned in the ODD+D protocol as a form of logic function which can be used to model agent behavior. Each of these will be reviewed in greater depth presently.

Selection Criteria

Criteria for evaluating the candidates for mapping states to actions had to be established. First and foremost, the state-action mapping must be implementable in a computer ABM, since that is the environment which has been identified as most appropriate for these types of experiments. Both candidates satisfy this criterion. Additional criteria came from those established through review of the ODD+D protocol, namely that the models make as few

assumptions as feasible, have sufficient justification for necessary assumptions, and be reasonably defensible.

Another criterion for selecting a method for mapping states to actions came from the lower-level process of experimentation: The method must facilitate exploration and analysis of alternatives. If the mapping cannot be easily modified then the process of evaluating alternatives may become unnecessarily difficult and increase costs to unacceptable levels. Furthermore, the manipulation of the model should have a theoretical or empirical backing, not be defined by trial-and-error, and be capable of generating alternatives outside the bounds of established knowledge.

Lastly, the state-action mapping must be amenable to the myriad behaviors which one may wish to model. Decisions made by agents can fall into one of two categories: Continuous or discrete. Examples of continuous decisions are orienting oneself and maneuvering in space. The decision to launch a missile is an example of a discrete one. It is important to note that every continuous decision can be discretized, albeit with a loss in resolution. It is more difficult, but not impossible, to map a continuous variable to a set of discrete actions. Ideally, the state-action mapping would be able to accommodate both types of behaviors.

Mathematical Functions

Optimal control theory relies heavily on the treatment of problems with pure mathematics. There are two types of mathematical models which can be used to map state observations to actions: Algebraic functions and transcendental functions. Algebraic functions are ones which involve "only a finite number of repetitions of addition, subtraction, multiplication, division, extraction or roots, and raising to powers" [4]. These elementary operations can be used to create functional models of behavior, e.g. in the form of (3.1) where $\mathbf{u}$ is the action, x_j are the elements of the n-dimensional state vector, and $k_{i,j}$ are constant coefficients.

$$u(\mathbf{x}) = \sum_{i=0}^{m} \sum_{j=0}^{n} k_{i,j} x_j^i \tag{3.1}$$

Transcendental functions extend algebraic ones in that the former do not have an exact representation as a finite sum of the latter. The trigonometric, exponential, and logarithm functions are examples of transcendental functions. They enable periodic and asymptotic responses which, while possible, would be more difficult to implement using only algebraic functions. Complex responses can be produced by algebraic and transcendental functions, and they can be combined to create even more sophisticated state-action mappings.

Algebraic functions might be poorly suited to problems of discrete decision-making. A potential work-around might be to apply a conditional logic function to the output, as in (3.2) where the threshold value α is used to control the action selection based on the evaluation of the algebraic function. However, this type of mapping falls apart in the general n-ary output case. Consider the ternary selection (3.3), where the pair of thresholds α_0 and α_1 are used to determine the action taken. The ordering of the discrete actions y_0, y_1, and y_2 becomes significant in this case, since switching the place of any pair would fundamentally change the decision-making process.

$$\mathbf{u} = \begin{cases} \mathbf{u}_0 & \text{if } u(\mathbf{x}) < \alpha \\ \mathbf{u}_1 & \text{otherwise} \end{cases} \tag{3.2}$$

$$\mathbf{u} = \begin{cases} \mathbf{u}_0 & \text{if } u(\mathbf{x}) < \alpha_0 \\ \mathbf{u}_1 & \text{if } \alpha_0 \leq u(\mathbf{x}) < \alpha_1 \\ \mathbf{u}_2 & \text{otherwise} \end{cases} \tag{3.3}$$

Decision Trees

Decision trees consist of sequences of logical gates (nodes) applied to input data which can terminate at any number of outputs (leaves) [113]. Each node necessarily branches into at least two paths, the end of which could be either another node with its own subsequent branches or a leaf. Conditional logic within each node evaluates part or all of the informa-

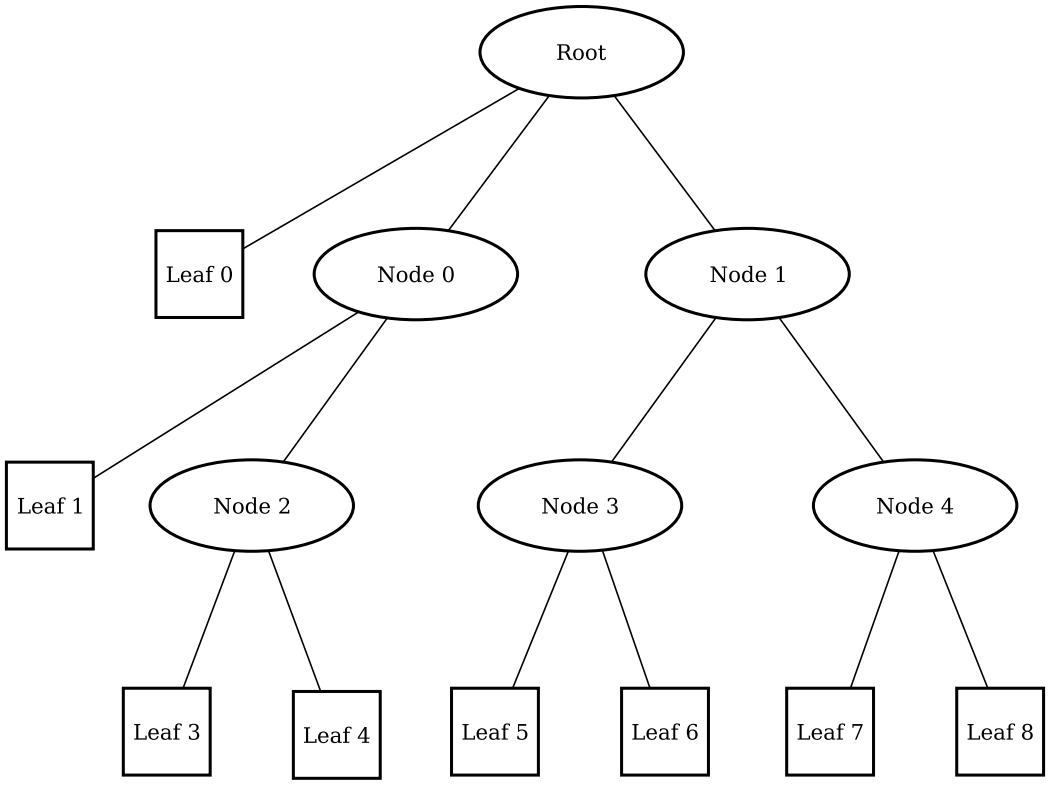


Figure 3.1: A notional decision tree

tion in the state vector to determine which branch to follow. Decision trees can simulate complex decision-making processes by passing the data through layers of logic. A notional decision tree is shown in Figure 3.1.

Decision trees can be easily implemented in a computer environment. The conditional logic in each node can be directly implemented with the if-else statements available in most programming languages. Tracing the logic which produced a decision at a given state can also be done easily. Consider the notional tree shown in Figure 3.1. It would be very straightforward to determine how the model arrived at, for example, Leaf 6: The logic would have had to follow [Root]→[Node 1]→[Node 3]→[Leaf 6].

Decision trees are also only capable of mapping to discrete action spaces. This poses a challenge to problems with continuous action spaces and could compound the difficulties identified in manipulating the logic within the tree.

3.2 Gap 3.2: Parametric Exploration Using Numerical Optimization

There are two dimensions to the space of state-action mappings. The first is the structural dimension, which takes different forms depending on the choice of apparatus selected. For mathematical functions, experiments along the structural dimension are concerned with the choice of terms to include in the model. Forward selection in linear regression is an example of structural experimentation with mathematical functions [20]. There are infinitely many possible structures for any mathematical function. This can be proven by the trivial example (3.4). It would therefore be necessary to impose an upper limit on the maximum allowable power $N \in \mathbb{N}$.

$$y = \sum_{n=0}^{\infty} k_n x^n \tag{3.4}$$

Structural experiments on decision trees would alter the number of nodes, order of nodes, and connections between nodes. However, these experiments would have to be conducted carefully so as to avoid changes which trivialize certain decision paths. For example, if the connection between [Node 1] and [Node 3] in Figure 3.1 were removed and a new connection between [Node 0] and [Node 3] created then [Node 1] would become irrelevant because it could only result in moving to [Node 4].

The other dimension to experimentation is a parametric one. For mathematical models, the parametric dimension regards the coefficients of the state variables used to determine the action. In (3.4), the values of the k_n are in the parametric dimension. The parameters of decision trees are the evaluations within each node which are used to select the path taken, such as whether to move to [Node 3] or [Node 4] from [Node 1].

The two dimensions are coupled for mathematical functions since a coefficient of zero is equivalent to omitting that term from the function. Similarly, the lack of a connection between two nodes in a decision tree can be viewed as a connection with an impossible criterion. This may allow the experimentation effort to focus solely on the parametric

dimension, as long as it may be assumed that all terms or connections which are potentially relevant have been included in the model.

Experiments in the parametric dimension of behavior modeling can be done using numerical techniques. Numerical optimization is a class of techniques for manipulating a set of parameters to improve the desirability of some metric function [96]. At a conceptual level, this is exactly what was sought by the third research question. The standard form of a constrained optimization problem is given by (3.5), where f is the function to be minimized, $\mathbf{x}$ is the vector of independent variables, g_i are the inequality constraints, and h_j are the equality constraints. The set of solutions to (3.5) are the vectors x^* given by (3.6), where $\hat{X}$ is the set of all feasible solutions.

$$\begin{aligned} \text{minimize } f(\mathbf{x}) \quad & w.r.t.\ \mathbf{x} \\ s.t. \quad & g_i(\mathbf{x}) \leq 0,\ i \in [1, n] \\ & h_j(\mathbf{x}) = 0,\ j \in [1, m] \end{aligned} \tag{3.5}$$

$$X^* = \{\mathbf{x}^* \in \hat{X}\ s.t.\ f(\mathbf{x}^*) \leq f(\mathbf{x})\ \forall \mathbf{x} \in \hat{X}\} \tag{3.6}$$

In the context of computer ABMs, the independent variables $\mathbf{x}$ are the parameters of the behavior model and f is the performance index stated in the fourth step of the controller construction process. The sets of inequality and equality constraints may be empty for certain problems, or may reflect limitations on the types of behaviors sought. Numerical optimization is an iterative process by which candidates for $\mathbf{x}^*$ are evaluated and compared to the current best solution $\hat{\mathbf{x}}^*$. The best-so-far is then updated, and the process repeated until convergence has been achieved or some other termination criteria as been met.

Techniques for numerical optimization fall into three main categories: Zeroth-order methods, which use only the value of the performance index f to determine the next set of $\mathbf{x}$ to be evaluated; first-order methods, which use the gradient $f'(\mathbf{x})$ or an estimate thereof to make an informed guess about how $\mathbf{x}$ should be modified to improve the performance

index; and second-order methods, which use the second derivative $f''(\mathbf{x})$ to achieve the same end. Second-order methods can be very expensive and so will be excluded from consideration here. A review of well-established and often-used methods will be given in the following sections, many of which would be applicable to both candidate state-action mapping techniques.

3.2.1 First Order Methods

First order methods for numerical optimization are so named because they make use of the objective function $\nabla_{\mathbf{x}} f(\mathbf{x})$ to determine how the candidate solution $\mathbf{x}$ should be updated in search of a better solution [52]. The update rule is given by (3.7), where α is the step size. The gradient is positive when the function increases as its argument increases. If the argument $\mathbf{x}$ is a vector then the vector of partial derivatives can be calculated with respect to each element independently.

$$\mathbf{x}_{k+1} = \mathbf{x}_k - \alpha \nabla_{\mathbf{x}} f(\mathbf{x}_k) \tag{3.7}$$

First order methods can be very powerful tools for optimization. The first derivative gives the direction of greatest change in the objective function with respect to the independent variables. However, calculating the gradient can be very costly. If the objective is not a simple function with an analytic derivative then an estimate of the gradient must be obtained e.g. using (3.8), where h is the step size. Obtaining such an estimate of the gradient becomes expensive as the dimensionality of $\mathbf{x}$ increases, in which case partial derivatives must be calculated by applying (3.8) to each element of $\mathbf{x}$ individually. An n-element vector of parameters requires $n + 1$ function evaluations to calculate all partial derivatives.

$$\nabla_{\mathbf{x}} f \approx \frac{f(\mathbf{x} + h) - f(\mathbf{x})}{h} \tag{3.8}$$

Estimating derivatives of the performance index with respect to the parameters of the

decision-making model in this way might be impractical if the simulations are expensive to run. Further, if random phenomena are included in the environment model then the multiple runs may be necessary to estimate the derivative, or the update rule might have be constrained to mitigate the potential for the algorithm to overshoot because of misestimation. Both cases would increase the computational cost to perform the optimization.

Improvements to the basic gradient descent algorithm can enhance its capabilities and applicability to more complex problems. Trust regions can be implemented to improve stability and prevent overshooting. Conjugation of the direction vectors used in the update rule can be retained as a kind of memory between iterations to expedite convergence. Lastly, gradient information can be retained between iterations to estimate the second derivative of the objective function, providing more information for the algorithm to operate on.

These methods have several drawbacks in the context of this research. Calculating derivatives of performance with respect to behavior model parameters may not be possible for decision trees for one of two reasons. First, small perturbations in the model parameter might have no effect on the path followed at the corresponding node, causing the estimated derivative to be zero. Second, if the decision *did* change as a result of the perturbation then the effect would be a discrete change in behavior, the derivative of which would be undefined. The calculations would likely be expensive for mathematical functions with many parameters because of the large number of simulations required to populate the vector of partial derivatives. Lastly, this approach is exploitative by definition and does not easily allow for explorations. This makes it susceptible to getting stuck in local optima, since the basic algorithm will not attempt to step "uphill" in search of a better solution. Exploration can only be reasonably achieved by running the algorithm from multiple starting points, which would increase the cost.

3.2.2 Zeroth Order Methods

Zeroth order methods, also called derivative-free methods, are iterative numerical optimization techniques designed to overcome some of the limitations of first order methods, particularly those associated with gradient calculation. As noted by Nocedal and Wright, the finite difference approximation (3.8) "cannot be regarded [as] a general-purpose technique ... because the number of function evaluations required can be excessive and the approach can be unreliable in the presence of noise" [96].

Derivative-free optimization can be performed in a variety of ways. It may be possible to create a surrogate model of the objective function using a handful of function evaluations – far fewer than would be required for an estimate of the gradient vector – and standard first order operations can be performed on the surrogate. For example, the parameters $\{\beta_i : i \in [0, n]\}$ of the linear model (3.9) can be estimated by solving (3.10), where X is the vector of sampled points in the design variable space and F is the vector of corresponding objective function values.

$$\hat{f} = \beta_0 + \beta_1 x_1 + \ldots + \beta_n x_n \tag{3.9}$$

$$\begin{bmatrix} \beta_0 \\ \beta_1 \\ \vdots \\ \beta_n \end{bmatrix} = (X^\top X)^{-1} X^\top F \tag{3.10}$$

Model-based methods for derivative-free optimization can be powerful if the objective function can be adequately approximated by the polynomial base. However, if the regressed model does not faithfully recreate the observed values, which can be checked using a residual or coefficient of determination, then the algorithm may struggle to converge quickly or reliably. Model-free techniques have been developed to mitigate the risk in estimating the

objective function.

Particle Swarm Optimization

A popular model-free, zeroth-order optimization technique is the particle swarm optimizer (PSO), which takes inspiration from the flocking behavior exhibited by animals in nature [124]. PSOs begin by initializing many unique candidates for $\mathbf{x}^*$. Each of those candidates is simulated and has its performance index calculated. The candidates are then updated according to some rules. The set of rules used to update models can vary by implementation but generally consists of three parts:

1. Convergence: The desire to maximize performance
2. Inertia: The tendency to maintain motion in a given direction
3. Separation: The desire to maximize the minimum distance between neighbors

These three rules can be combined into a "velocity" term describing how the combination of independent variables associated with each member changes over time. An example expression for velocity for the i^{th} member of a population is given by (3.11), where $\vec{v}$ is the particle velocity, α, β, and γ are constants, $\tilde{\mathbf{x}}$ is the vector of independent variables corresponding to the member with the highest performance, and $\mathbf{x}_{closest}$ is that corresponding to the nearest neighbor. The independent variables of the member are then updated according to (3.12), where dt is the step size.

$$\vec{v}_i(t) = \alpha(\tilde{\mathbf{x}}(t) - \mathbf{x}_i(t)) + \beta\vec{v}_i(t-1) + \gamma(\mathbf{x}_i(t) - \mathbf{x}_{closest}(t)) \tag{3.11}$$

$$\mathbf{x}_i(t+1) = \mathbf{x}_i(t) + \vec{v}_i(t)dt \tag{3.12}$$

PSOs are easy to implement and can be easily tuned to solve a wide variety of problems. They can incur high costs, especially for large swarms, and that can hinder broad

explorations of the parameter space. However, if the computational cost is acceptable then a PSO can explore large portions of the parameter space and potentially identify multiple local optima if implemented correctly.

Genetic Algorithms

Another popular zeroth-order method is the genetic algorithm (GA) [157]. Like PSOs, GAs are bio-inspired methods which emulate the process of natural selection to explore and exploit the design space. This is done by representing points in the space as binary strings – an analogy to chromosomes in biology. An example is given in (3.13). The modeler must specify the precision of the binary conversion via the number of bits to use. A mapping between space may also be necessary since binary can only be used to represent integers. An example of this is shown by the underbraces in (3.13), where the base 10 integer represented by each binary string is given and must be divided by 10^5 to decode the actual parameter value.

$$\mathbf{x} = \{3.14159, 2.71828\} \rightarrow [\underbrace{1001100101100101111}_{=314159}\underbrace{100010010111010100}_{=271828}] \tag{3.13}$$

Another approach would be to map the binary string to the unit interval, and then map the unit interval to a range of values the parameter may take. However, this places a limit on the maximum and minimum values which can be explored which may be undesirable. Furthermore, the binary representation is necessarily discrete and so limits the algorithms capacity to fully explore the space of alternatives. More bits can be added to increase the resolution but that would come at an additional cost, and the number of bits required may not be known a priori.

The setup for this method is largely identical to that for a PSO, with the main difference being the use of a binary representation for design variables. Binary strings are

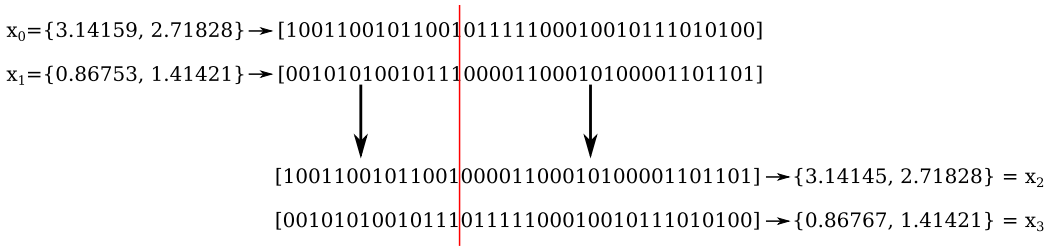


Figure 3.2: Example of binary chromosome crossover in genetic algorithms

necessary for the two main methods by which a GA generates new candidates for evaluation: Crossover and mutation. Crossover is a process by which two chromosomes are intermixed to generate two new ones. This can be done via one-point crossover, where a bit index is selected at random and both chromosomes are split at that point. New chromosomes are created by combining e.g. the bits preceding the split from the first chromosome with the bits after the split from the second. This is depicted in Figure 3.2, where the red line indicates the index where the split occurs. Two-point crossover is also possible, where splits occur at two points and the bits between them are swapped.

Like PSOs, GAs can have high computational costs because of the number of function evaluations required at each iteration. The limitations on precision imposed by the binary representation may also be undesirable. Further, the binary representation necessarily constrains the space of parameters which can be explored. This would be highly undesirable since the "best" values for any decision-making parameter to take are likely to be unknowable at the outset.

3.3 Gap 4: Engagement Analysis

Analyzing the interactions between entities working together or against one another is the subject of an established field: Game theory. The mathematical bases of game theory were established by von Neumann and Morgenstern in 1944 [94]. The primary concern of game theoretic analyses is to gain deeper insights into "complex situations where two or more individuals ... can choose among a set of available options" [23] and explore "the ways in

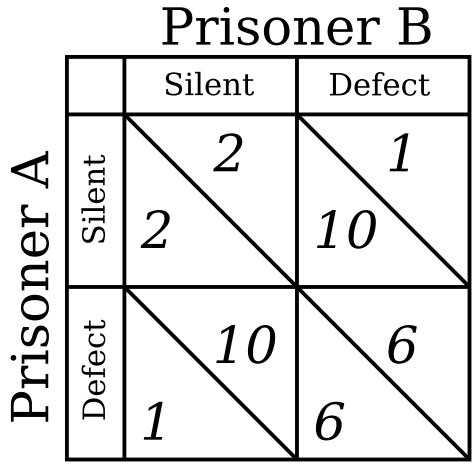


Figure 3.3: A version of the Prisoner's Dilemma game

which interacting choices of ... agents produce outcomes with respect to the preferences (or utilities) of those agents" [111].

Game theory poses a simple question: What is an agent's best course of action? This question is predicated on expectations of other agents' courses of action, as well as their expectations. It is the expectations that make analyses through game theory difficult. The prisoner's dilemma (PD) is the classic example used to introduce concepts in game theory. In PD, two prisoners are being interrogated separately. Each is given a choice: They can defect, implicating the other and potentially reducing their penalty; or they can remain silent. If both choose to defect then they both receive an increased penalty, while if both remain silent then both received a reduced penalty. This is often depicted in the tabular form shown in Figure 3.3, where the values shown are penalties and lower is more preferable. Either prisoner would reason that the best overall strategy would be to remain silent. However, they could reduce their penalty by defecting if the other remained silent. Similarly, if the other defects but they remain silent then they receive a significantly increased penalty. Thus, the stable solution is for each to defect, resulting in a sub-optimal outcome overall.

The stable solution of joint defection is known as a Nash equilibrium, named after John Forbes Nash who provided several important existence proofs for such equilibria points and greatly advanced the seminal work of von Neumann and Morgenstern [90]. A Nash

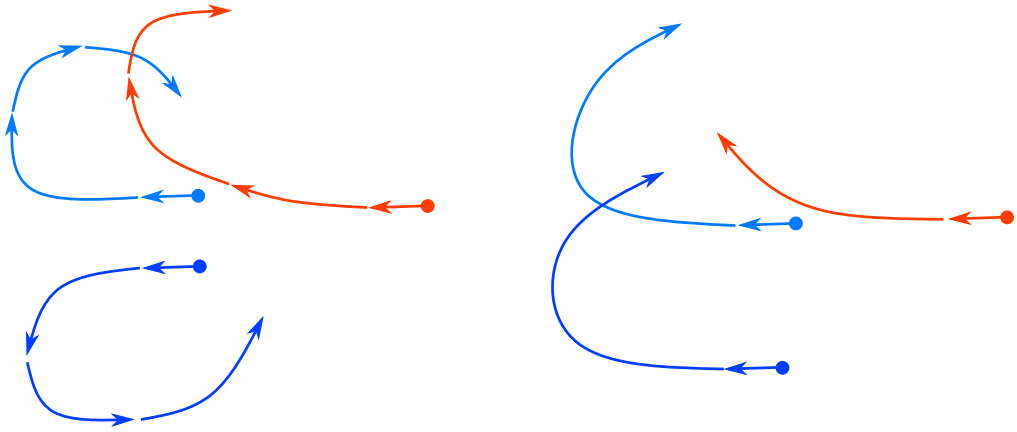
equilibrium prescribes the strategies of each player in the game such that none can achieve a better outcome by unilaterally deviating. That is, it represents the best option for each player assuming all others do not change their strategies.

Nash proved the existence of equilibrium points in games. However, there can be *multiple* such equilibria for a single game, finding any one of them can prove mathematically intractable, and knowing how many there are may be impossible [23]. In this way, the PD game is a trivialized example which is useful for conveying basic concepts, but fails to capture some deeper intricacies of game theoretic analysis.

An extension of the base PD game can be used to aid in understanding these deeper concepts: The Iterated Prisoner's Dilemma (IPD). The IPD is a simple extension of the PD, where the same game takes place several times in succession. This modification can increase the complexity of analyses significantly. For one, the number of possible game trajectories increases exponentially and analyzing every possible path quickly becomes infeasible: There are 4^n possible paths for an IPD game with n repetitions, and there are more possible paths after 40 steps than the estimated number of stars in the observable universe [143]. Several factors can influence the types of strategies which form Nash equilibria for repeated games, such as: how many rounds are played and whether or not the players are aware of such; whether or not the players have perfect information about the game state; and whether or not any player in the game acts with uncertainty.

Applying Game Theory to The Research Objective

Game theory provides an important perspective on the behavior modeling problem for scenarios with multiple interacting agents. The two-on-one engagements shown in Figure 2.7 can be viewed as a game, where the turning and extending fighters are cooperating with one another and competing against the aggressor. The maneuvers shown are only two of infinitely many possible trajectories the game can follow, each of which involves a slightly different combinations of decisions made by each player. Shaw described several other sec-



(a) Defensive split *(b) Sandwich*

Figure 3.4: Other possible section tactics, adapted from [123]

tion tactics which have the same initial condition as the half-split, including the defensive split and sandwich tactics shown in Figure 3.4.

The variety of section tactics available in literature forces a reconsideration of the behavior modeling problem. The scope of the effort must attempt to include considerations for the behaviors of *all* agents within the model. Failure to do so could jeopardize the integrity of the model. However, this dramatically increases the complexity of the problem. The emphasis shifts from optimizing a single performance metric to finding a balance between multiple, related, and possibly competing objectives. Furthermore, the curse of dimensionality becomes increasingly difficult to address as the number of agents, and thus the number of possible trajectories, increases. Lastly, the temporal credit assignment problem becomes a more-general credit assignment problem, where modifications to an agent's behavior model must consider the effects of other agents' actions on outcomes.

Reformulation of the Behavior Modeling Problem

Adopting a game-theoretic perspective of behavior model construction would be necessary to mitigate the risks in experimenting with alternative employment concepts, despite the increased complexity it brings with it. Assuming the behaviors of other agents would

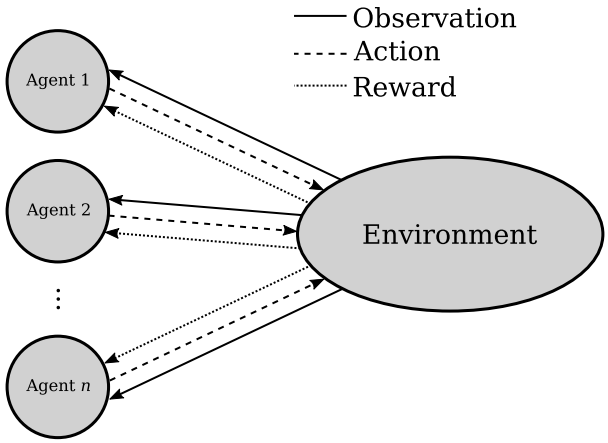


Figure 3.5: Agent-environment interactions for multi-agent systems

not deviate from some preconceived idea would violate several risks identified by Alberts and Hayes, namely that explorations would be confined to well-established borders and creativity would not be adequately captured. A reformulation of the behavior modeling problem was therefore necessary.

Firstly, the agent-environment interaction diagram shown in Figure 2.3 must be extended to explicitly identify all agents in the model. The result is Figure 3.5, where each agent is viewed as observing and interacting with the environment separately. The change is relatively minor: Each agents' perspective of the environment is largely identical to the single-agent case, since the other agents are effectively a part of the environment. However, each agent is capable to influencing the environment through its actions, and other agents are able to perceive and be subjected to that influence. The explicit separation of each agent from the environment when viewed from a higher level establishes a basis for the rest of the problem reformulation.

3.3.1 Multi-Objective Optimization

The optimization problem statement (3.5) no longer directly applies to the behavior modeling problem in general because of possible competition between agents with diametrically opposed objective functions. That is, the behavior model construction process for each

agent would be attempting to solve a version of (3.5) which is specific to that agent and conditioned on the behavior models of every other relevant agent. For example, in the two-on-one section tactics described by Shaw, the fighters are attempting to minimize the likelihood they will lose the engagement while the aggressor attempts to maximize it. The behavior of each agent is distinct for each of the four tactics shown, and the details of each are dependent upon those of the others. They are, ostensibly, highly effective tactics but beg the question as to whether or not other tactics might be equally or more effective.

The theory of numerical optimization provides a mechanism for addressing problems with multiple competing objectives, aptly named multi-objective optimization [108]. The purpose of multi-objective optimization differs slightly from that of single-objective optimization: A set of solutions is sought, rather than a single, point solution. The set of solutions is characterized by the concept of dominance. A candidate solution $\mathbf{x}_0$ is strongly dominated if there exists another candidate $\mathbf{x}_1$ such that (3.14) holds, where $\{f_i(\cdot)\ \forall\ i \in [1,n]\}$ is the set of objective functions and minimization of each is assumed to be the goal.

$$\mathbf{x}_1 \succ \mathbf{x}_0 \iff f_i(\mathbf{x}_1) < f_i(\mathbf{x}_0) \forall\ i \in [1,n] \tag{3.14}$$

The PSO and GA algorithms each have variations which can accommodate multi-objective problems. A popular and effective GA is the Non-dominated Sorting Genetic Algorithm II, which uses dominance levels in place of the fitness metric in single-objective GA [35]. The set of non-dominated solutions found through application of an appropriate optimization algorithm is known as the Pareto frontier, an example of which is shown in Figure 3.6 for the Binh-Korn test function [17]. A point on the Pareto frontier is a possible solution to the optimization where no metric can be unilaterally improved. This conforms to the notion of a Nash equilibrium in game theory. Indeed, the set of strategies constituting a Nash equilibrium *must* be non-dominated [111].

Multi-objective optimization algorithms have several drawbacks which may hinder their use in this research. The first is that the size of the parameter space being optimized over

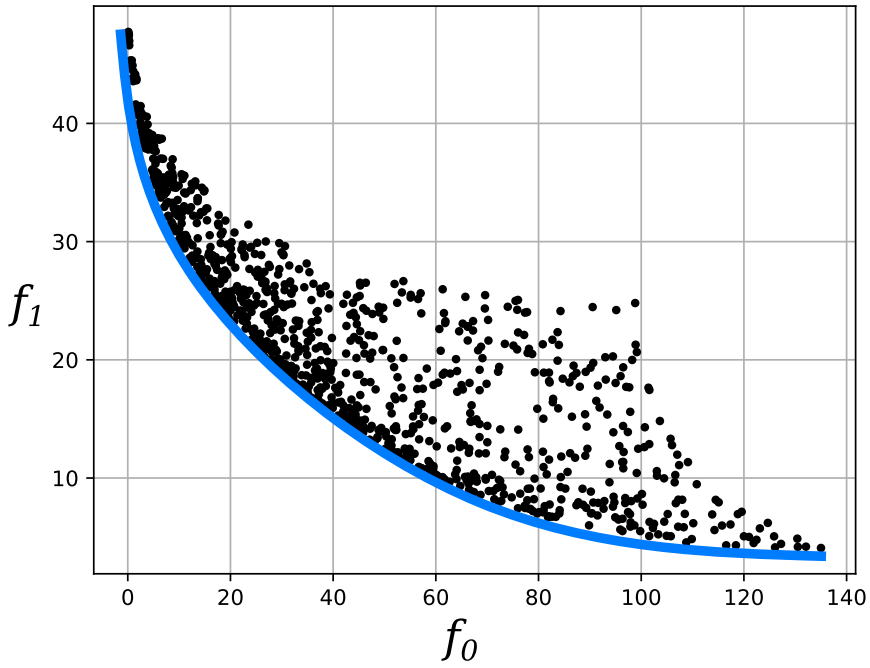


Figure 3.6: The Pareto frontier (blue line) for the Binh-Korn test function [17]

would become very large, as the parameters for each agent must be joined into a single vector. Second, the size of the objective space would also become high-dimensional and the Pareto frontier could become discontinuous or non-smooth. Lastly, multi-objective optimizers provide a set of solutions, but the ultimate choice of which solution to go with is left to the user. Selecting any subset of solutions for further analysis might compromise the integrity of that later analysis. Doing so may also risk imposing artificial constraints on the process, or lead to the conclusion that explorations were unnecessary.

3.3.2 Multidisciplinary Design Optimization

Multidisciplinary design optimization (MDO) is a class of techniques for solving complex design problems involving multiple interrelated subproblems or disciplines, where "the performance of the multidisciplinary system is driven not only by the performance of the individual disciplines but also by their interactions" [80]. Optimizing the aerostructural analysis problem shown in Figure 3.7 is a good example of the types of problems which

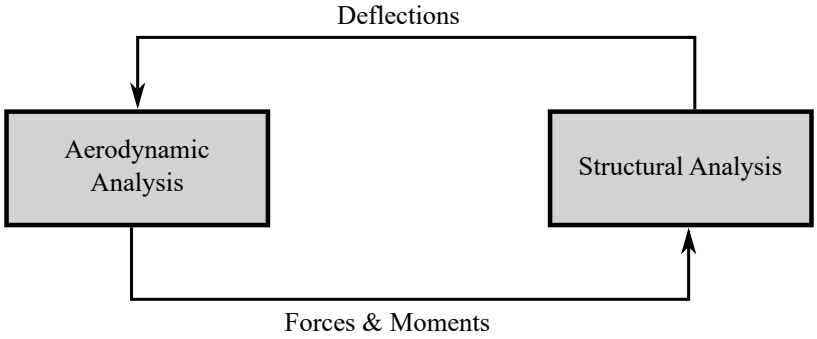


Figure 3.7: Coupling between disciplines in aerostructural analysis

require MDO: Aerodynamic forces influence structural optimizations which, in turn, alter aerodynamic characteristics. These two disciplines have to solved simultaneously because of the coupling between them.

A key concept in MDO is that of feasibility. A disciplinary solution is feasible if "the equations the discipline code is intended to solve are satisfied" [31]. Solutions will vary across disciplines: A feasible aerodynamic design does not guarantee a feasible structural one, nor vice versa. This leads to two distinct classes of MDO algorithm: Multidisciplinary feasible and individual discipline feasible. Multidisciplinary feasibility is achieved when the various disciplines have achieved feasibility *and* the inputs to each are feasible solutions to all other, related disciplines. Individual discipline feasibility is achieved when feasibility is realized within each discipline but *not* between them.

MDO algorithms wrap around the disciplinary analyses to solve a variation of the optimization problem (3.5). For example, MDO can be used to optimize the internal structure of a wing subject to a given flight condition and outer mold line. The optimizer could perturb the position, thickness, and number of ribs and spars to reduce weight while producing the necessary lift and not exceeding conditions for mechanical failure.

Multi-Agent Behaviors as a Multidisciplinary Problem

The multi-agent behavior modeling problem could be viewed as a type of multidisciplinary problem. This would allow the large, multi-agent optimization problem to be decomposed

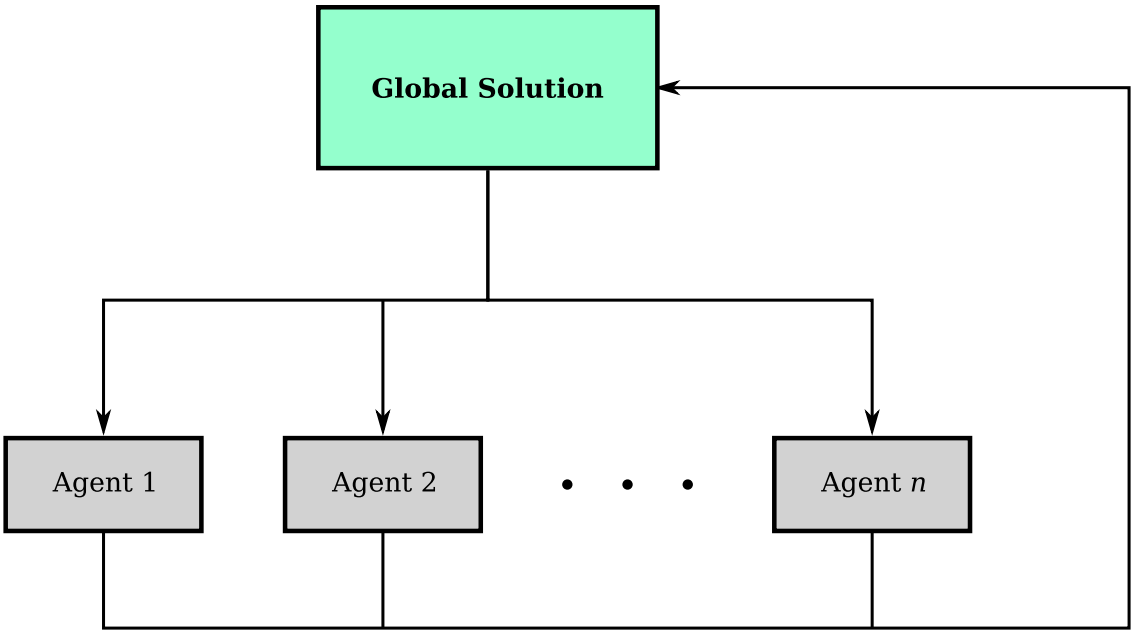


Figure 3.8: Notional multi-agent solution approach inspired by multidisciplinary design optimization

into several distinct problems which can be solved individually. Each agent in the model would be attempting to solve its own version of (3.5), subject to the partial solutions – i.e. behaviors – provided by every other agent. This might look something like Figure 3.8, where each agent takes steps towards optimizing its behavior based on the current best solution produced by the other $n - 1$ agents. A global repository of behaviors is then updated with these partial solutions, and passed down for the next iteration.

The MDO-inspired approach has several potential benefits. First, massively parallel computing capabilities can be leveraged to solve the small, individual optimization problems in parallel. This can dramatically reduce the time required to perform the analyses. Second, it simplifies the optimization problem because it is only being solved at the agent level, rather than at the level of all agents simultaneously. That is, each agent is searching for a solution to the question: *Given how all other agents are behaving, how should I behave to as to maximize my expected performance and effectiveness?* This may make the problem more tractable because the behaviors of other agents are assumed, and the resulting interactions can be explored more thoroughly. The iterative nature of the approach

would allow for those assumptions to change as the optimizer progressed.

There are potential drawbacks to using this approach. There is potential for the couplings between agents to result in a stiff system, where perturbations in the behavior of a single agent can dramatically alter the quality of other solutions. This would require the optimizers to take extremely small steps so as not to overshoot and end up in an infeasible region of their respective spaces. This relates to another potential problem: The conditioning of the individual optimizers on the solutions provided by others might inhibit thorough explorations. Multiple optimizations may have to be run from different starting points to adequately explore the high-dimensional space. For example, optimizing for the two variations in the half-split maneuver shown in Figure 2.7 might require the implementation of two distinct optimizers, one for cases where the aggressor engages the turning fighter and the other for cases where it engages the extending fighter. If the choice of which fighter to pursue were left to the aggressor model then it may get stuck in a local optimum corresponding to one of the two cases and never explore the other.

3.4 Gap 1: Design Attributes

Any potential solution to the first gap, concerning the consideration of design attributes in the exploration of alternative employment concepts, would likely depend on how the other, lower-level gaps were addressed. The targeted literature search for this gap had to be performed last because of this.

As discussed throughout this dissertation, the processes of innovating on tactical fronts have largely relied on human input. Unfortunately, this also means literature on how design attributes might impact employment concepts is non-existent; potential techniques appear to reside solely in the minds of operators and analysts [92]. However, a closer examination of the problem and inspiration from Biltgen's SOCRATES provide some insights into how this gap can be addressed.

Altering a design attribute of a system will likely affect how that system interacts with

its environment: F-35 pilots can engage with tactics that might be considered risky for other systems because of its advance design attributes. There are several possible approaches to creating agent decision-making models in a way which considers design attributes. The choice of approach may depend on how much variation in behavior is expected with respect to variations in the design attributes.

At one extreme, a unique process of experimentation with employment concepts could be instantiated for every point in the sampling of design attributes. This would obviate the need to include design attributes in the decision-making processes explicitly, and the optimization could proceed without any modifications. However, it is clear from previous findings that this would be impractical at best, and impossible at worst, because of the large number of samples often considered in DSE.

At another extreme would be experimentation to produce *robust* models of behavior. Robustness is "the sensitivity of empirical results to credible changes in model specification" [160]. In the context of the present discussion, the "credible changes" are the variations in design attributes. This approach would be focused on producing singular models of behavior which maximize expected performance irrespective of the design attributes. This could simplify the training process by treating the sampling of design attributes as random variations – i.e. noise – in the environment. However, there could be no exploitation of technological advantage because the models would not be explicitly aware of them.

It may be reasonable to expect neighboring points in the design space to perform similarly under identical employment concepts. However, this assumption would not be reasonable for more distal design points. It may, then, be possible to conduct experiments on alternative employment concepts within smaller regions of the design space. However, the appropriate partitioning of the design space may not be knowable a priori, necessitating some meta-experimentation to determine the most suitable discretization.

Biltgen used augmentation of the state space to facilitate regression of expected performance as a function of design attributes. That is, his method treated design attributes as

additional observable states. This allowed agents to select different courses of action from their respective sets based on current design attributes. However, those courses of action were prescribed and so did not fully allow the agents to capitalize on the potential benefits of their technologies. An analogous method to could be adopted, where the lower-level behavior models at the center of this research are given an expanded state space which includes sampled design attributes. This could allow the numerical optimization procedures to tune the behavior model to different combinations of design attributes. However, it could also dramatically increase the cost to perform the optimization if the model complexity would have to be increased to accommodate the larger state space. That is, there would be more parameters to manipulate.

3.5 Summary of Findings

A theoretical basis upon which explorations of alternative employment concepts was found in optimal control theory. A generic process for constructing controllers was distilled from three sources of literature. Closer examination of the process showed the primary challenges resided in a single step: Experimentation with alternatives.

Experimentation requires an apparatus and process. In the case of this research, the apparatus is the model of behavior which maps stimuli to responses or, analogous, state observations to admissible actions. The process of experimentation prescribes the manipulations of those mappings used to produce desirable models. Two techniques for mapping states to actions were found in literature on optimal control theory and agent-based modeling: mathematical functions and decision trees, respectively. Three methods for exploring the parametric dimension of behavior modeling were identified from literature on numerical optimization: particle swarm, genetic algorithm, and gradient descent. Some incompatibilities between candidates for each were found, but both candidates for mapping states to actions could be subjected to at least one of the techniques for exploration.

Critical challenges were identified in the exploration techniques. First, it was shown

that neither mathematical functions nor decision trees could accommodate both continuous and discrete action spaces. Decision trees might have a slight advantage on this basis because any continuous space can be discretized. However, increasing the number of discretizations increases the number of possible decision paths, which would exacerbate the curse of dimensionality.

A significant drawback to using either mathematical functions or decisions trees to map states to actions is their rigid structure. The modeler must specify the terms to include or connections between nodes a priori, and determining the minimum viable subset may be difficult at the onset of the modeling process. The modeler may choose to err on the side of caution and include more terms or connections and allow the optimizer to prune the model by driving unnecessary parameters to zero, but that could increase the cost to perform the optimization. Furthermore, the optimizer may fail to prune the model effectively, which can lead to overfitting [61] and may undermine confidence in the results.

Existing methods for numerical optimization may not adequately address the temporal credit assignment problem. The single objective optimization problem (3.5) condenses the effects of all decisions and interactions into a one metric, making it impossible to separate the effects of different decisions on the outcome. Exploring all possible trajectories in this way may be impossible because of the curse of dimensionality. This challenge is made more significant from the perspective of multi-agent systems, where the curse of dimensionality and credit assignment problem are multiplied and, therefore, greatly exacerbated.

Multi-objective and multidisciplinary design optimization techniques may be feasible approaches to resolving the difficulties of multi-agent problems. However, these methods may fall short in terms of the breadth of explorations into alternative employment concepts which is possible. Multiple instances of the optimizer may have to be run in order to adequately cover the space, increasing the computational cost of such methods.

Lastly, it was found that very little literature was available on the incorporation of design attributes into the experimental process for behavior model construction. Three possible

Characteristic \ Alternative	1	2	3
Gap 3.1 State-Action Map	Mathematical Function	Decision Tree	
Gap 3.2 Experimentation & Exploration	First Order Optimization	Zeroth Order Optimization	
Gap 4 Engagement-Level Analyses	Multi-Objective	Multi-Disciplinary	
Gap 1 DSE	Partitioned	Robust	Parametric

Figure 3.9: Morphological matrix of candidate solutions to the identified gaps

techniques were identified through a decomposition of the problem: Partitioning the design space and creating models for specific regions, creating models which are robust to design attributes, and parameterizing the models by including design attributes in the state space.

The alternatives for each gap are captured in the morphological matrix shown in Figure 3.9. There are 30 compatible solutions, the lone incompatibility being the use of gradient descent on decision trees. It would be impractical to test every possible combination to determine the most effective approach to the research objective. Furthermore, the structural issues of both mathematical functions and decision trees poses a significant challenge to the use of any alternative. The persistence of this gap in the existing techniques motivated a broader search.

3.5.1 An Alternative Approach to Behavior Exploration

Artificial neural networks (ANNs) are a type of model inspired by biology, providing a general method for function representation in computational sciences [60]. ANNs have gained recognition for their ability to accurately approximate a wide variety of functions

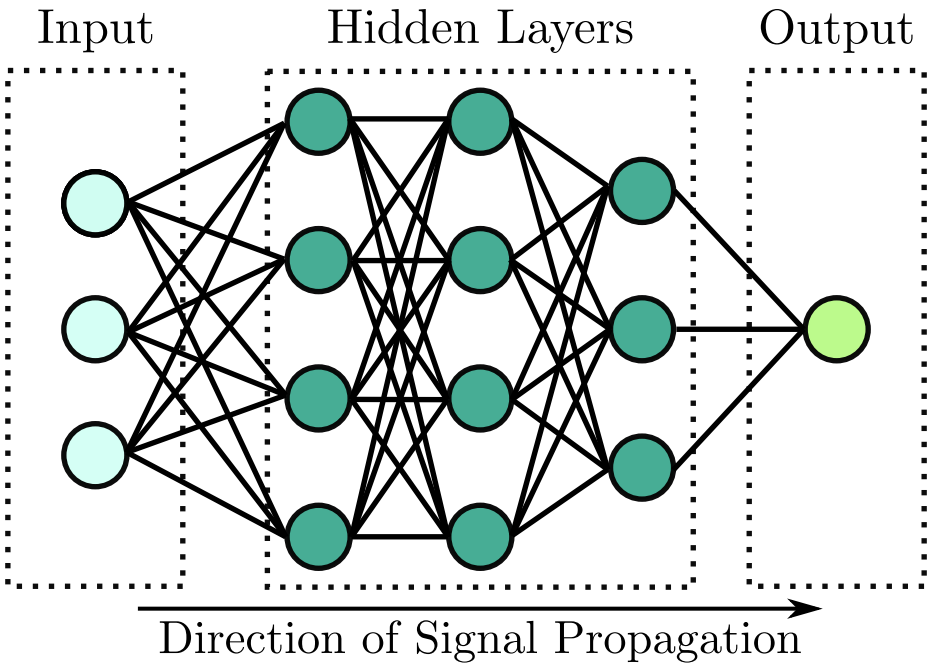


Figure 3.10: Notional artificial neural network architecture

without the need to assume the form of said functions. They can handle both discrete and continuous variables in both the input or output sides.

Machine learning (ML) is a broad class of techniques for creating models for a variety of purposes. ML can be used for non-linear regression and classification tasks. ML techniques prescribe the modification of model parameters to improve its performance in some metric space. The model in question can be, though is not necessarily, an ANN.

A notional ANN is shown in Figure 3.10. Inputs to the model are passed through at least one hidden layer, each applying a transformation to the input vector and passing the result to the next layer. The general form of the transformation applied to the inputs is given by (3.15), where ϕ is an arbitrary activation function, b is a constant bias, w_i are scalar weights, and x_i are the input values. This functional form is very similar to that of algebraic functions; the power of ANNs comes from the activation function ϕ, which is typically non-linear, and the use of multiple hidden layers.

$$y = \phi\left(b + \sum_{i=1}^{n} w_i x_i\right) \tag{3.15}$$

Any continuous function can be approximated by an ANN with at least one hidden layer of arbitrary width [32]. This suggests an ANN could be used as a substitute for algebraic functions in any situation where the latter would be appropriate. ANNs can also have multiple outputs, and selection algorithm can be implemented over those outputs to produce a capability which is similar to that afforded by decision trees. The process of creating and refining these approximations is known as *training* the ANN, and there are several training methods available in the literature [67]. These methods are based on optimization theory and therefore meet the standards established by the ODD+D protocol.

Constructing and training ANNs poses several challenges which have yet to be resolved by the communities which use them. There are two general issues concerning the use of ANNs. First is the selection of a network topology – how many layers the model has, referred to as model "depth"; how many nodes each layer should have, referred to as model "breadth"; and how the layers should be connected to one another. This is typically left to heuristics and trial-and-error rather than theory [158]. The second issue is in setting the hyperparameters used to control the training process. The number of updates to the model, amount of data used, and the learning rate are examples of hyperparameters which must be specified. Some guidance exists on tuning hyperparameters or using search methods to find effective values, but those values may be unknowable a priori [131].

Several methods for manipulating the parameters of ANNs are available in literature. Classes methods span several different purposes for ANNs, including function approximation, classification, and policy optimization. The existence of these methods in the literature is a significant benefit to the use of ANNs. However, while the mathematics behind these methods are sound, explaining *why* the model produced a certain output may prove difficult, if not impossible [14].

There are three classes of ML: supervised learning, unsupervised learning, and reinforcement learning. Supervised learning is a form of regression using ANNs, where the goal is to minimize the error in the prediction made by the ANN on a set of training data

which consists of paired inputs and outputs [33]. The supervised learning problem is: Given a set of n inputs X with corresponding outputs Y, find the parameters of the ANN θ such that the prediction error $\frac{1}{n}(\hat{Y} - Y)^2$ is minimal. In the context of this research, supervised learning would be useful in cases where the desired action was known for all possible states and the modeler sought to implement that mapping. However, this would not be the case in exploratory analyses of tactical alternatives in the system design process, so supervised learning would not be appropriate.

Unsupervised learning seeks to create models of data which can be used for such tasks as outlier detection, classification, or data compression [49]. None of these capabilities would be particularly useful in the context of behavior exploration.

Reinforcement learning (RL) is "learning what to do – how to map situations to actions – so as to maximize a numerical reward signal" [138]. It can be described as "a way of programming agents by reward and punishment" [67]. Its formulations draw inspiration from psychology, as evident by the language used to describe it, making it an appealing candidate for this work. The goal of RL is for agents to learn incrementally better behaviors through interactions with their environment. Mathematical techniques may be used to estimate a measure of utility over the action space, and the model parameters updated accordingly. These elements of the RL paradigm make it a good fit for the problem of behavior exploration in the context of this research.

Algorithms for Reinforcement Learning

There are several algorithms within the umbrella of RL. One of the earliest examples was the tabular method of Q-learning developed by Watkins [156]. Q-learning seeks to populate a table where each row corresponds to an observable state and each column to a possible action. Each entry in the table is a numeric value indicating the "value" of the state-action pair indicated by the row and column. Rewards obtained through simulation are used to update the tabulated values according to (3.16). In this equation, $Q(s, a)$ is the current

value in the table, α is the learning rate, r_t is the reward for selecting action a_t in state s_t, $\gamma \in [0, 1)$ is a discount factor.

$$Q(s_t, a_t) \leftarrow Q(s_t, a_t) + \alpha \left(r_t + \gamma \max_{a_{t+1}} Q(s_{t+1}, a_{t+1}) - Q(s_t, a_t) \right) \tag{3.16}$$

Strong parallels can be drawn between solutions to the update rule (3.16) and the performance index (**??**), repeated below for convenience. Suppose the function $Q(s, a)$ were known exactly for all s and a, such that $r + \gamma Q(s', a') - Q(s, a) = 0$. Then (3.17) would hold by induction. Furthermore, if $\gamma = 1$ then $Q(s_t, a_t)$ is equivalent to the performance index J calculated from that point forward.

$$J = S(\mathbf{x}(N)) + \sum_{k=0}^{N-1} L(\mathbf{x}(k), \mathbf{u}(k)) \qquad (\textbf{??}, \text{repeated})$$

$$\begin{aligned} Q(s_t, a_t) &= r_t + \gamma \max_{a_{t+1}} Q(s_{t+1}, a_{t+1}) \\ &= r_t + \gamma(r_{t+1} + \gamma \max_{a_{t+2}} Q(s_{t+2}, a_{t+2})) \\ &= \gamma^N r_{t+N} + \sum_{i=0}^{N-1} \gamma^i r_{t+i} \end{aligned} \tag{3.17}$$

Reinforcement learning using a method like Q-learning can help to mitigate the temporal credit assignment problem; that was a significant part of what motivated its development. The discount factor γ weights future rewards based on how temporally distant they are from the present. This has two effects. First, it allows rewards to propagate backwards in time, and allows the model to develop associations between actions and potentially delayed outcomes. Second, it encourages the model to explore alternative paths far away from epicenters of high reward, where those rewards would have low weights.

Tabular Q-learning can be effective for problems with small state and action spaces. However, large state spaces can consume large amounts of computer memory and make

the approach intractable. ANNs can be used to mitigate this issue by approximating the unknown Q function. That is, instead of a table, an ANN is used to predict the values of all actions given an input state vector. The model is trained using (3.16) with minor modifications and essentially reduces to a problem of supervised learning [85].

Q-learning is a value method: The model is not learning how to select actions, only the value of making those selections. Action selection can be done in two ways: greedy and ε-greedy. The greedy approach has the agent select the action with the highest predicted value, exploiting its knowledge of the state-action space to pick the trajectory with the best reward. The ε-greedy approach infuses the agent with a penchant for exploration. An agent using ε-greedy selects the highest-valued action with probability $(1 - \varepsilon)$ and another, random action with probability ε. This allows the agent to deviate from what it thinks is the best behavior and to explore the state-action space more thoroughly.

Value methods like Q-learning were found to be theoretically intractable for certain classes of problems [139]. This led to the development of policy methods, which extended RL by explicitly representing the action selection as a function to be optimized. The goal becomes finding the parameters of the action selection function – i.e., the policy – which maximize the expected reward, where that expectation comes from a learned value function. The value function used for policy methods is often very similar, if not identical to the Q-function.

There are several policy methods used in literature. Deep deterministic policy gradient methods were used by Lillicrap et al. and Silver et al. to achieve high levels of performance on a variety of problems [74, 126]. Trust region policy optimization was developed to address issues of stability which can plague RL implementations [119]. The method takes its name from a standard optimization technique whereby the optimizer is penalized for deviating too far from the previous solution. This helps to ensure the model is always improving but does overshoot when updating its parameters. Proximal policy optimization succeeded trust region policy optimization by virtue of being easier to implement while

maintaining its capacity to achieve high levels of performance on benchmark problems [120].

Proximal policy optimization (PPO) was developed by Schulman et al. as a data-efficient and reliable RL algorithm with low complexity compared to other state-of-the-art techniques. As a policy method, the goal of PPO is to find the function $\pi^* : \mathbb{R}^n \rightarrow \mathbb{R}^m$ which maximizes the expected reward, where n is the number of observable states and m the number of admissible actions. Each output from the function corresponds to the probability of taking that action. This probability mass function (PMF) is sampled using a categorical algorithm to determine what action the agent will take.

The PMF is tuned through training such that less-favorable actions are assigned lower probabilities and more-favorable ones are assigned higher probabilities. The use of a stochastic process for action selection as two primary benefits. First, it allows the model to act in ways which are expected to be sub-optimal based on current information, which can greatly enhance explorations of alternative behaviors. Second, it agrees with the conceptual model of behavior established through behavioral psychology. Skinner observed that behaviors are non-deterministic: Presented with the same stimulus on multiple occasions, a single organism often exhibits a variety of behaviors rather than a single, predictable, and repeatable one [130].

PPO updates the parameters of an ANN using an estimate of the advantage function. The advantage A^π of action a_t given state s_t is given by (3.18) [118]. This formulation reduces the variance in the estimate of the advantage function, which can help with stability and data efficiency during training. In practice, the advantage function will not be known exactly and so will have to be estimated. Schulman et al. showed that this function can be estimated using (3.19), where the estimated value of the current state $\hat{V}(s_t)$ is deducted from the discounted sum of all subsequent rewards. The value function can be estimated

using a separate ANN trained to solve a regression problem.

$$\begin{aligned} A^{\pi}(s_t, a_t) &= Q^{\pi}(s_t, a_t) - V^{\pi}(s_t) \\ &= \mathbb{E}_{\substack{s_{t+1:\infty} \\ a_{t:\infty}}} \left[\sum_{\tau=0}^{\infty} \gamma^{\tau} r_{t+\tau} \right] - \mathbb{E}_{\substack{s_{t+1:\infty} \\ a_{t+1:\infty}}} \left[\sum_{\tau=0}^{\infty} \gamma^{\tau} r_{t+\tau} \right] \end{aligned} \tag{3.18}$$

$$\hat{A}^{\pi}(s_t, a_t) = -\hat{V}(s_t) + \sum_{\tau=0}^{\infty} \gamma^{\tau} r_{t+\tau} \tag{3.19}$$

The advantage estimate $\hat{A}^{\pi}(s_t, a_t)$ is used to update the parameter of the ANN in PPO by stochastic gradient descent. An estimate of the gradient of the reward with respect to the ANN weights and biases θ can be obtained using (3.20), where $\log\left(\pi_{\theta}(a_t|s_t)\right.$ is the log probability of selecting action a_t given state s_t, subject to the current policy parameters. These gradients can be difficult to calculate because the set of parameters θ can be very large. Algorithms for automatic differentiation have been developed to facilitate training with these methods [102].

$$\hat{g} = \mathbb{E}\left[\hat{A}^{\pi} \nabla_{\theta} \log\left(\pi_{\theta}(a_t|s_t)\right) \right] \tag{3.20}$$

A step of the gradient descent optimizer operating on (3.20) will attempt to increase the probabilities of actions with high advantages and decrease the probabilities of those with low advantages. This becomes apparent when constructing the loss function whose gradient is (3.20), namely (3.21). Issues can arise if the estimate is significantly different from the true value, which can cause the optimizer to take too big a step. That is, training can destabilize if the policy deviates too much in a single iteration. The authors of PPO initially sought to resolve this by creating the "surrogate" objective (3.22), which uses the ratio of probabilities from the current policy θ and past policy θ_{old} to discourage large changes in the PMF, and the Kullback-Leibler divergence KL is included as an additional

implicit constraint [72].

$$L^{PG}(\theta) = \mathbb{E}\left[\hat{A}^{\pi}\log\left(\pi_{\theta}(a_t|s_t)\right)\right] \tag{3.21}$$

$$L^{PG}(\theta) = \mathbb{E}\left[\frac{\pi_{\theta}(a_t|s_t)}{\pi_{\theta_{old}}(a_t|s_t)}\hat{A}^{\pi} - \beta KL[\pi_{\theta_{old}}(\cdot|s_t), \pi_{\theta}(\cdot|s_t)]\right] \tag{3.22}$$

The objective function (3.22) proved to be difficult in practical applications. PPO uses a simple clipping mechanism instead of a divergence metric to restrict the policy update step. The new objective function is given by (3.23), where $\mathcal{R}_t(\theta) = \frac{\pi_{\theta}(a_t|s_t)}{\pi_{\theta_{old}}(a_t|s_t)}$ and ϵ is a small positive number [120]. This new objective function disincentives large policy changes, effectively stabilizing the update while simplifying the implementation.

$$L^{CLIP}(\theta) = \mathbb{E}\left[\min\left(\mathcal{R}_t(\theta)\hat{A}_t, \text{clip}\left(\mathcal{R}_t(\theta), 1-\epsilon, 1+\epsilon\right)\hat{A}_t\right)\right] \tag{3.23}$$

Applications of Reinforcement Learning

Reinforcement learning has been used to produce a variety of interesting results, particularly when it comes to playing arcade games [85]. Arcade games are well-suited to the RL paradigm: The model observes the state of the game, selects a control signal such as a maneuver or actuation command, and receives a reward in the form of a score. The goal of an arcade game is often to achieve the highest score possible using the limited set of controls available to manipulate and interact with the environment. While these results are undoubtedly significant in the advance of the academic body of knowledge pertaining to RL, the simplicity of arcade games and the rules they abide by do not adequately substantiate the use of RL in approaching more complex problems.

It has also been shown that applications of RL can produce models capable of achieving super-human levels of performance on more complex problems than arcade games. A notable example of this was the work by Silver et al., who were able to train an ANN

to outperform human experts in the game of Go [127]. This was significant because, on a standard 19×19 board, the game of Go has been estimated to have more than 10^{170} legal position [145]. Identifying all legal positions has proven to be a monumental task, estimated to take roughly 10,000 weeks for a 6×6 board in 1994. This makes any attempt at exploring all sequences of legal positions to identify truly optimal strategies decidedly infeasible.

The complexity of the game of Go cannot be dismissed, but it could be argued that it pales in comparison to other problems. By observation, Go is a discrete game; the environment can only present one of a finite, albeit quite large, number of possible states, and players can only select one of a finite number of discrete actions to take per ply. A game with a continuous state or action space, or both, would be infinitely more complex than Go from this perspective. However, RL has been successfully applied to problems even more complex than Go.

Vinyals et al. trained an ANN to play the real-time strategy computer game StarCraft II [153]. Their model was capable of outperforming human players in competitive scenarios using a combination of supervised and reinforcement learning techniques. Baker et al. trained competing ANNs to play a game of team hide and seek [11]. Their models discovered how to use tools to achieve their goals and appeared to have developed a basic model of foresight and preparation. Zhang et al. used RL to train an agent engaging in an air-to-ground combat scenario, and demonstrated effective learning on two variations of the scenario with different levels of complexity in the environment [162]. Pope et al. developed an effective model for maneuvering a fighter aircraft using RL [105]. Their model achieved a record of five wins and zero losses against an expert human pilot in simulated combat.

Multi-Agent Reinforcement Learning

RL has been successfully applied to problems with multiple interacting agents [153, 11, 105]. The formulation of the multi-agent reinforcement learning (MARL) problem scarcely differs from the single-agent case. Rather, MARL forms a kind of *autocurriculum*, whereby any changes to the behavior of one agent alter the experiences of those which interact with it and, therefore, influence their learning processes. Dynamic interactions allow agents to explore regions of their respective behavior spaces simultaneously, which has the potential to produce models which are robust to variations in the behaviors of other agents.

MARL is not without its drawbacks. Learning in multi-agent scenarios may take longer, since the models have to interact with and adapt to a changing environment. This compounds the increased computational cost to train multiple ANNs simultaneously. Parallel computing can alleviate some of this burden. However, this leads to another issue: It may be desirable to train multiple models *per agent* in order to better explore the behavior space. Training a single model per agent would capture a single trajectory through the combined state-action spaces, which can be immense. Multiple models may be required in order to ensure the space has been adequately explored, as well as to mitigate the risk of models falling into local minima or being exploited.

Baker et al. implemented a method of "decentralized execution and centralized training" [11]. Each agent is given a copy of a "master" behavior model and allowed to interact with their own instance of the environment, which includes other agents whose models are similarly copied form the global scope. The experiences of each agent are then gathered at training time, and a degree of omniscience is allowed at this time, since the agents can have imperfect information about their environment during the simulation. Decentralized execution allows the model to sample a broader array of state observations and, hopefully, to develop a more robust policy as a result. Centralizing the update step increases sample efficiency by dedicating all resources to improving a single model. However, the authors note that using an entirely decentralized setup, where each agent is given a unique model,

can achieve comparable levels of performance.

Incorporating Design Attributes

Literature on the use of RL in the context of DSE is extremely limited. However, the possible approaches to enabling considerations of design attributes for ANNs trained with RL are the same as before: Partitioning the design space, creating robust models, or augmenting the state space. Biltgen demonstrated the inclusion of design attributes in the model state space with his SOCRATES methodology, where doing so allowed the ANN to regress expected performance as a function thereof. However, this example differs slightly from what is being attempted here and therefore does not constitute substantive evidence in support of the method.

Challenges in Reinforcement Learning

Reinforcement learning is not a silver bullet. Training ANNs can take a considerable amount of time: the StarCraft II model took over 40 days to train, and the hide and seek models required several hundred million episodes of training. These immense costs could be argued as justifiable because of the exploratory and bias-mitigating aspects of RL. Advances in hardware, such as training on graphics processing units and massive parallelism, have made training more feasible, but the costs remain high [10].

Another potential challenge to using ANNs is their notoriety for being opaque and difficult to interpret [14]. The mappings between states and actions performed by an ANN can be difficult, if not impossible to explain because of the layered, non-linear transformations used. Explainable artificial intelligence is an active, albeit nascent area of research [36]. This is especially problematic for multi-agent problems, where interactions between models can further obfuscate the causes of certain effects.

CHAPTER 4
HYPOTHESIS COMPOSITION AND TESTING

> *"All of our science courses tell us the way to solve a problem is to break it into smaller parts and analyze the parts. And that has been phenomenally successful for every branch of science. But the great frontier in science today is, what happens when you try to go back, to put the parts together to understand the whole?"*
>
> — Steven Strogatz

The objective of this research was to develop a new methodology for enabling employment concept exploration in system design. A basic methodology was identified via literature search, but several gaps were identified which had to be addressed. This established several research questions, which were the focus of the targeted literature searches presented previous chapter. This chapter will synthesize those findings to establish a new methodology to fulfill the research objective.

4.1 Synthesis of a New Methodology

A more-detailed methodology was synthesized from what was found in the available literature on design space exploration, modeling and simulation, and optimal control theory. The new methodology is shown in Figure 4.1, where exploration of employment concepts is explicitly included in the analysis process. Relevant questions from the ODD+D protocol will be identified for each step.

4.2 Steps 1-4: Defining the Problem Space

The first three steps of the process are primarily concerned with defining the problem space to be analyzed. Step 1 establishes the purpose of the effort, such as exploring possible

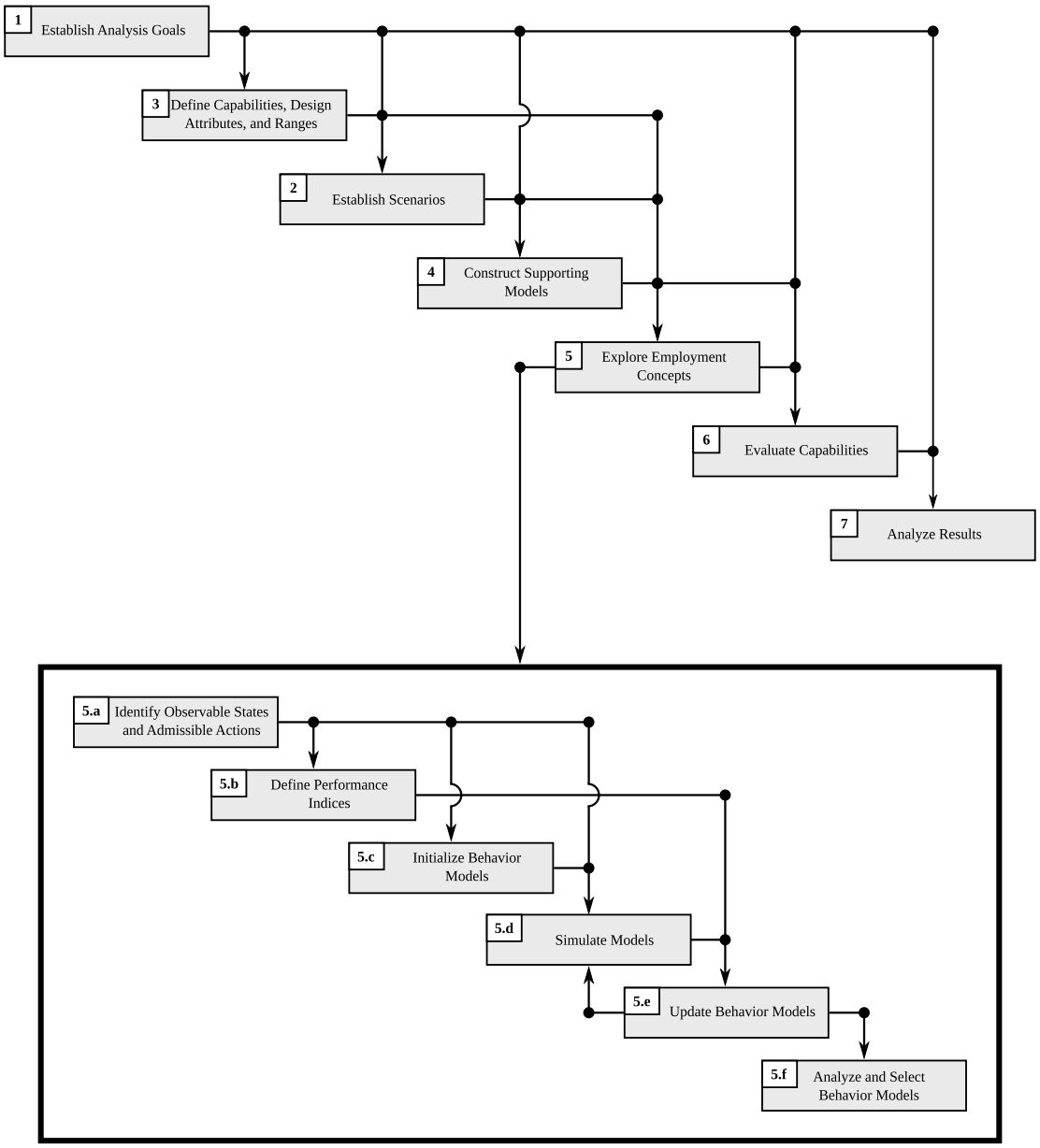


Figure 4.1: Proposed methodology

solutions to an existing or expected capability gap and the associated MOEs and MOPs. This step will answer the protocol questions *(I.i.a) What is the purpose of the study?* and *(I.i.b) For whom is the model designed?*

Step 2 defines the capabilities, design attributes, and ranges to be assessed. Characteristics of evolutionary or transformational solutions to capability gaps will be the main drivers of this step. A morphological matrix can be created to facilitate this step, and the design attributes corresponding to the selected morphology can be derived by mapping technologies to k-factors. Few, if any, protocol questions would be directly addressed by this step. This is largely because the ODD+D protocol was not created for design space exploration.

Step 3 defines the scenario or scenarios of interest. This step is largely carried over from the basic process without modification. The output of this step will be a conceptual model of the environment and agents within in. This step will address the protocol question *(I.ii.a) What kinds of entities are in the model?*

Step 4 is to construct the necessary models in an appropriate computing environment using the agent-based modeling paradigm. Models of agent motion through the environment, interactions between agents, or communications are constructed in this step. If necessary, surrogate modeling can be used to enable rapid evaluation of costly models to facilitate the analysis effort [16]. The ODD+D protocol applies most directly to this step in the process. The entirety of the *Design Concepts* and *Details* sections, less those questions pertaining to agent decision-making processes, should be addressed here.

4.3 Step 5: Exploring Employment Concepts

Step 5 constitutes the primary contribution of this research. As shown in Figure 4.1, explorations of employment concepts follows its own methodology which is contained within the larger process. This sub-methodology follows the generic controller construction process distilled from optimal control theory, where the experimentation step is decomposed into the initialization, simulation, and update steps from the learning process inspired by

behavior psychology.

4.3.1 Outlining Behaviors: Steps 5.a & 5.b

Step 5.a is to identify the observable states and admissible actions for each agent. This will answer several questions from the ODD+D protocol, including:

(I.iii.a) What entity does what, and in what order?

(II.i.b) On what assumptions is/are the agents' decision model(s) based?

(II.ii.f) Do spatial aspects play a role in the decision process?

(II.ii.g) Do spatial aspects play a role in the decision process?

(II.iv.d) Are the mechanisms by which agents obtain information modelled explicitly, or are individuals simply assumed to know these variables?

Step 5.b in this sub-methodology is to define the performance index or indices for each agent. The MOEs and MOPs defined at the onset of the effort will come into play here, and must be translated into quantitative metrics for implementation and calculation in the computer environment. The performance index (**??**) should be utilized whenever possible to ensure adherence to theory. This requires the MOEs and MOPs to be defined in terms of the observable states and admissible actions to facilitate the exploration of employment concepts. This step will also answer the protocol question *(II.x.a) What data are collected from the ABM for testing, understanding and analysing it, and how and when are they collected?*

4.3.2 Mapping States to Actions: Step 5.c

Step 5.c is to initialize the behavior models. The mapping between observable states and admissible actions will be defined during this step. Technical challenges in this step were identified through problem decomposition, resulting in the identification of Gap 3.2 and formulation of Research Question 2.

Three techniques were identified for addressing the gap: Mathematical functions, decision trees, and artificial neural networks. It was noted that both mathematical functions and decision trees impose significant limitations on the experimentation process because of their rigid structure. ANNs are general function approximators and so would not be expected to have the same limitations, possibly enhancing the exploration process by removing the influence of structural limitations in the state-action mapping. However, available information was insufficient to dismiss these techniques altogether, leading to the statement of the first hypothesis:

Hypothesis 1

If artificial neural networks are used to map observable states to admissible actions then broader explorations of employment concepts will be possible because the models will not be constrained by structural limitations

4.3.3 Simulation: Step 5.d

The next step in the sub-methodology is to generate quantitative data on the performance and effectiveness of the behavior models through simulation of their interactions with their environment and, where applicable, one another. Simulation of the environment model typically involves solving equations by stepping through time. Behavior models will be queried at appropriate times to allow agents to act. The states observed, actions performed, and corresponding performance measures defined in Step 5.b must be tracked throughout the simulation to enable the next step.

4.3.4 Updating Behavior Models: Step 5.e

Step 5.e uses the data collected through simulation to determine how the behavior models should be changed, if at all, in an attempt to realize higher performance and greater ef-

fectiveness. The curse of dimensionality and credit assignment problems are of particular concern here, since the number of actions taken likely represents only a small slice of all possible decision paths and attributing credit or blame to any single action can become very difficult, if not impossible. These observations formed the basis of Gap 3.2 and Research Question 3.

Techniques for numerical optimization were explored in relation to this step. Particle swarm optimization, genetic algorithms, and gradient descent were identified as possible solutions to the gaps. However, closer inspection of the optimization problem indicated these standard methods might not adequately address the temporal credit assignment problem because they operate on single objectives, which confound the effects of individual actions by aggregating performance into a singular metric. This could prevent the update step from preserving the "good" behaviors in favor of extinguishing the "bad" ones. Reinforcement learning was identified as a potential solution which *does* consider the effects of individual actions on overall performance. Furthermore, RL has a basis in operant conditioning and has been successfully applied to a variety of problems. However, applications of RL to design space exploration had not been found in literature. This led to the statement of the second hypothesis:

Hypothesis 2

If reinforcement learning is used to train artificial neural networks then effective exploration of employment concepts will be possible because individual actions will be considered, mitigating the credit assignment problems

Multi-Agent Considerations

Gap 4 and Research Question 4 were derived from observations on the challenges associated with the potential for interactions between agents to influence the quantitative mea-

sures produced by Step 5.d, and those influences could affect this step. The temporal credit assignment problem becomes a more general credit assignment problem here; instead of temporal delays being the source of difficulty, it is the potential for outcomes to be effected by other agents that poses the most significant challenge. Any improvements to the agent's behavior model must consider the question: *How much did my actions contribute to my performance, and how much did the actions of others contribute?*

Two potential techniques for mitigating the challenges in this step were identified from literature on numerical optimization: Multi-objective optimization and multidisciplinary design optimization. Concerns were raised with respect to the increased problem complexity when using multi-objective optimization. MDO presents its own challenges: Interactions between agents might be tightly coupled, meaning any changes to one could significantly alter the evaluations of others. This could inhibit explorations if the models need to be improved very gradually to prevent destabilization. Models could also end up in a local optimum and have difficulty getting out of it. There may also be higher computational costs associated with MDO.

Multi-agent reinforcement learning was identified as a potential solution to Research Question 4. In MARL, agents are trained in environments alongside one another and experience evolving interactions. Multiple models can be created for each agent, and random sampling can be used to form groups which enhance the diversity of adversary and/or ally models each agent interacts with. This forms an *autocurriculum*, where any changes an agent makes to its behaviors are experienced by others, who alter their own behaviors in response. This allows each agent to be trained individually, like MDO, but can also allow for broader explorations of employment concepts, like multi-objective optimization. However, no examples of MARL being used in design space exploration were found in literature. These observations culminated in the third hypothesis:

Hypothesis 3

If multi-agent reinforcement learning with multiple models per agent is used to train interacting agents in an engagement scenario then those models will be able to learn effective behaviors because the more diverse autocurriculum will enable broader exploration

Design Attribute Considerations

The last gap which had to be addressed concerned how considerations for variations in design attributes could be included in the exploration of employment concepts. The literature available on this subject was very limited, but three possible approaches were identified through closer inspection of the problem: Partitioned design space, robust models, and augmented state spaces. Creating robust models would be easy to implement because no changes to the behavior models would be necessary. However, it would make the models unaware of the design attributes and therefore unable to leverage them in the decision-making processes. Partitioning the design space and exploring employment concepts in those smaller regions might enable greater specialization but would increase computational costs. Augmenting the state space could enable greater specialization than would be possible with partitioning the design space while maintaining the lower cost of the robust model approach. This led to the statement of the fourth hypothesis:

Hypothesis 4

If design attributes are treated as observable states then the trained behavior models will be better able to mitigate or capitalize on different settings because the design attributes will be factored into the decision-making processes

Iterative Improvement

Steps 5.d and 5.e are to be repeated until satisfactory performance is achieved, adequate exploration has been conducted, or computational resources have been exhausted. It is common practice in RL literature to define the number of iterations, also called episodes, at the onset of the training process. There is, however, no theoretical limit on the amount of training which can be done, nor is there a method for determining how many iterations would be needed. Examples in literature have used as few as 5×10^5 episodes, and as many as 5×10^8. The number of iterations performed will likely depend on the availability and capabilities of computational resources.

4.3.5 Selecting Behavior Models for Evaluation: Step 5.f

It would be ideal if, for each agent, there was a single model which performed best in all possible environments and under all possible combinations of design attributes. However, evaluating every possible combination of environment and design attributes would likely be impossible. Furthermore, it is likely that, with the models made available by the sub-methodology, some models will perform better or worse than others under certain conditions and the analyst will be forced to choose a subset from among them to carry forward. Data collected for training the models could be used to estimate expected performance, e.g. the average performance index over the last 1,000 iterations. This would allow the analyst to identify and select those which achieved the highest performance during training, and omit those with the lowest performance.

All of the top-performing models could be carried forward for evaluation if available resources permitted so. Alternatively, a small DOE could be performed over the design space and a multi-attribute decision making (MADM) technique could be used further refine the model selection. The purpose of MADM techniques is to inform "preference decisions ... over the available alternatives that are characterized by multiple, usually conflicting, attributes" [64]. In this case, the attributes would be the measures of performance

and effectiveness under the conditions specified by the DOE, which would be assumed to be representative of the technologies being considered. A MADM technique, such as the Technique for Order Preference by Similarity to Ideal Solution (TOPSIS) or Simple Additive Weighting (SAW), could then be applied [107].

4.4 Steps 6 & 7: Evaluation and Analysis

The last steps in the main methodology would be straightforward and could be addressed using methods from literature. The behavior models resulting from Step 5 are simulated again in Step 6, and the relevant data are collected. If individual technologies with known effects on design attributes are being considered then the corresponding values can be implemented in the state space. If specific technologies have not been identified or their precise effects on design attributes determined then a DOE can be generated over the design space for evaluation, and the sampled values can be used to augment the state space. In either case, the quantitative performance data can be collected from the simulations to facilitate the next step.

Step 7 is where analysis of alternatives and design space exploration truly occur. If individual technologies were evaluated then their effects on the MOEs and MOPs can be inspected directly, and a determination can be made as to whether or not a subset of technologies, and their corresponding employment concepts, closes the capability gaps. If the goal was design space exploration then a DOE can be generated over the design space and each point simulated. The data produced by those simulations would enable the generation of surrogate models which could be used to interrogate the design space and identify regions of high performance or trade-offs between attributes.

4.5 Description of Experiments

> *"If you want to study something and you cannot prove a lot of things, it is good to start with something simple."*
>
> — Lior Bary-Soroker

The hypothesized solutions to the methodological gaps had to be tested in order to substantiate or refute the claims. It would not be reasonable to apply the entire methodology directly to a relevant problem since the underlying hypotheses constitute significant deviations from the status quo. Instead, a suitably representative problem had to be identified for testing the constituent hypotheses first. The experimental plan shown in Figure 4.2 was designed to build up the methodology from the bottom up. First, the M&S techniques would be tested, followed by considerations for dynamic interactions between entities, and the DSE aspects of the problem would be considered last.

4.5.1 Selection and Design of an Experimental Apparatus

A computational tool for ABMS was needed in order to perform experiments, based on the stated conjecture to the first research question. There existed a number of such tools in use across engineering disciplines, as well as more basic tools to create tailored ABMS environments. One had to be chosen from to meet the needs of the experimental plan.

Both scenarios – pursuit-evasion and air defense – require models for motion in three dimensions, models for sensing other agents, and the ability to handle large-scale simulations with many agents. The overall methodology requires the tool to allow intrusive modification of agent decision making models. Ideally, the tool would be: publicly available, current, and standardized; require low to moderate model development effort; and have low run times.

These criteria were applied to the ABMS tools surveyed by Abar et al. in 2007. Among those surveyed, 18 fit the criteria for development effort and scale requirements while only eight were listed as appropriate for military combat, war fighting, or air defense scenar-

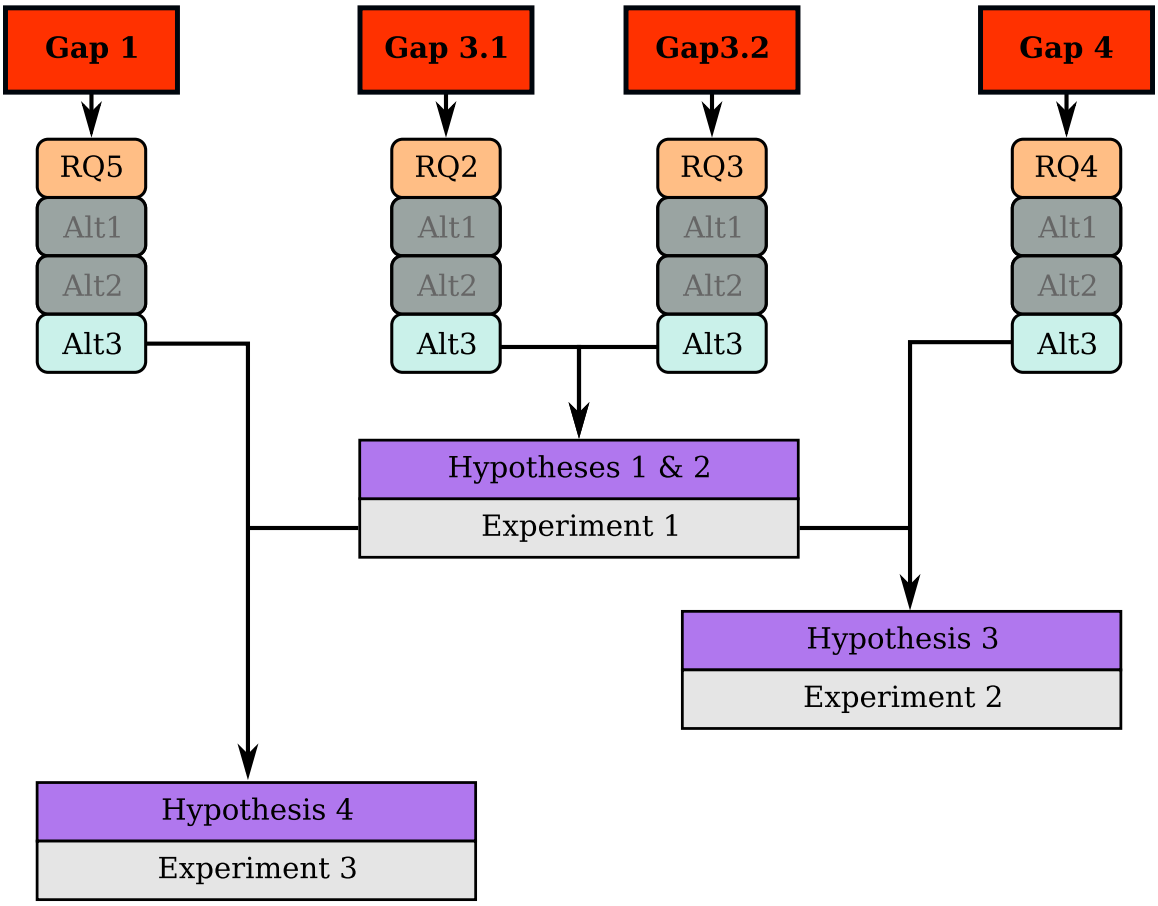


Figure 4.2: Diagram of experimental plan showing relationships between identified gaps, research questions, alternative solutions, hypotheses, and experiments

ios. The overlap between the two categories contained two entries: SimEvents and Simio. SimEvents is a package provided by MathWorks for use with the MATLAB software, which is not free and is generally regarded as being quite slow. Furthermore, the tool is closed source so intrusions to include adaptive behaviors may be challenging. Simio, also a closed source tool, is a framework for modeling intelligent objects. It uses a graphical interface to build processes for simulation, which is not desired here.

Abar et al. list another category which includes in its scope "Evolutionary computation or genetic programming, Artificial intelligence, Neural networks, [and] Robotics" [1]. Within this category is the Flexible Large-scale Agent-based Modeling Environment (FLAME). FLAME has been used to model conflict resolution among large numbers of agents [110]. However, the remaining literature using FLAME appears limited.

This tool should not be confused with the Flexible Analysis Modeling and Exercise System (FLAMES) used by Biltgen in his dissertation. FLAMES is a framework for building and simulating ABMs leveraging object-oriented programming. As demonstrated by Biltgen, FLAMES can be used to model adaptive behaviors in military operations. It has basic models of military assets to facilitate development, including those for motion, sensing, and communication [16].

Another framework has emerged recently as the US Air Force standard for M&S: The Advanced Framework for Simulation, Integration, and Modeling (AFSIM). It is "a government-approved C++ simulation framework for use in constructing engagement and mission-level analytic simulations for the Operations Analysis community, as well as virtual experimentation" [28]. The simulation engine is currently restricted by the International Traffic in Arms Regulations and so is only available to US DoD contractors and select academic institutions, though the results obtained from the engine are not necessarily controlled. However, it contains numerous sub-models for easily and rapidly constructing complex multi-agent scenarios. Multiple levels of fidelity are provided for sensing, motion, and communication. AFSIM provides its own scripting language to enable customization.

The last option is to build an ABMS tool from scratch. Any object-oriented programming language could serve as the basis for this development, such as Python, MATLAB, Java, or C++. Building a custom ABMS environment offers the ultimate flexibility, allowing motion, sensing, and communication models to be built as needed and providing the easiest route to behavior modeling. Notably, both Python and MATLAB have sophisticated machine learning add-on packages. The additional flexibility offered by taking this approach comes at the cost of required effort and trustworthiness. Building everything from scratch will be time consuming, and any model can be manipulated to spit out the desired results. Steps must be taken to ensure the tool is properly built and not biased in any way.

The objective of this research did not include the development of a new ABMS environment. The behavior modeling aspects were intended to be agnostic of the underlying simulation engine. That is, a general methodology for behavior adaptation should be independent of the environment and context of those behaviors. However, the existence and availability of powerful ML and RL capabilities in Python made it an appealing option for the conduct of experiments. Python can also wrap other ABMS tools, allowing its capabilities to be extended to other modeling tools and environments with relative ease. As noted, care had to be taken to ensure the models were developed properly.

4.5.2 Scenario Selection

An appropriate scenario was needed for application of the proposed methodology. This scenario had to satisfy two criteria, derived from the characteristics of the general problem of design space exploration on engagement-level system analyses. The criteria were:

1.a Multiple interacting agents with related objectives

1.b Performance of an agent's behavior is affected by changes in design attributes

Further criteria were derived from the experimentation process itself, the need to be able to falsify the hypotheses, and the resources available:

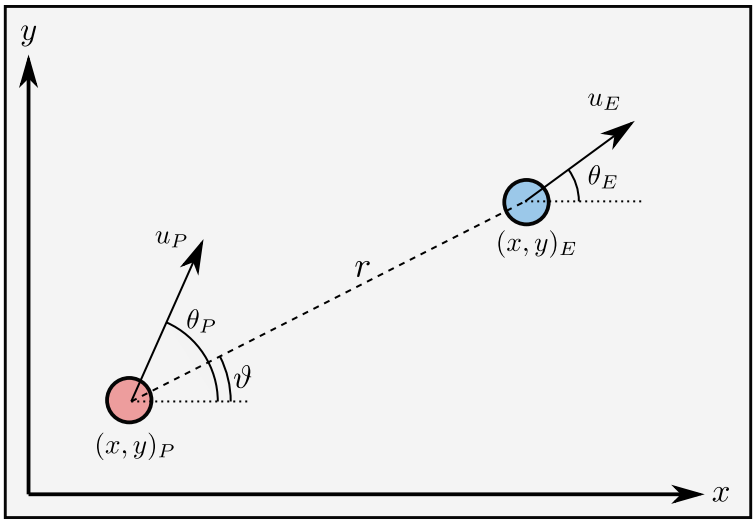


Figure 4.3: Geometry of the pursuit-evasion game

2.a Clearly defined measures of performance and effectiveness

2.b Examples of behaviors exist in literature and are accessible

2.c Low computational cost

Exposition

Two scenarios were considered in the course of the experimentation effort. The first was the game of pursuit and evasion, the description of which is given below and depicted in Figure 4.3.

The Pursuit-Evasion Game

Consider two entities, a pursuer P and an evader E. Both entities exist in a two-dimensional plane defined by coordinates x and y; move at constant speeds u_P and u_E, respectively; and can control their headings θ_P and θ_E, respectively. The objective of P is to capture E as quickly as possible, and the objective of E is to avoid capture for as long as possible.

This problem was selected for experimentation because, despite its apparent simplicity, it has several key features in common with more complex problems within the scope of the research objective. Games of pursuit and evasion require each of the two players to select an action at regular intervals to maximize their expected performance. However, each player must also consider the strategies employed by the other, and it is the interaction between the two competing strategies which dictates the outcome. The MOEs and MOPs are also easily defined: The pursuer is effective if and only if it is able to capture the evader, and performance can be measured in terms of the time elapsed until capture is achieved. The MOEs and MOPs of the evader are exactly opposite those of the evader. These features satisfy criteria **1.a** and **2.a**.

The pursuit-evasion game also satisfies criterion **2.b** for the pursuer. Two pursuit algorithms are readily available in literature: Pure pursuit (PP) [30] and proportional navigation (PN) [88]. The pure pursuit algorithm operates a very simple premise: The pursuer must eventually intercept the target if it is always pointing directly at it. Proportional navigation is more sophisticated, relying on exact knowledge of inertial velocities to calculate the lateral accelerations required to minimize miss distance. The equations for PP and PN are given by (4.1) and (4.2), respectively, where the heading rate $\dot{\theta_P}$ is taken positive counter-clockwise from the positive x-axis and K is the constant navigation gain.

$$\dot{\theta_P} = K(\vartheta - \theta_P) \tag{4.1}$$

$$\dot{\theta_P} = K\left(\angle\left(\vec{u}_P + \frac{\vec{r}\times(\vec{u}_E - \vec{u}_P)}{\|\vec{r}\|^2} \times \vec{u}_P\right) - \theta_P\right) \tag{4.2}$$

Evasion algorithms were more difficult to find. Jinking is provably optimal strategy against a pursuer using PN. However, executing this strategy requires "perfectly timed hard turns to the left and to the right" [50]. Solving the necessary equations to satisfy this criterion would be non-trivial. Instead, two simpler evasion strategies were derived from

first principles. The first was pure evasion (PE), which is essentially the opposite of PP: The evader attempts to minimize the closure rate of the pursuer, and therefore delay capture, by pointing directly away from the pursuer. The second is marginally more sophisticated: Beam evasion attempts to maximize the line-of-sight (LOS) angle rate $\dot{\vartheta}$ by maintaining a velocity perpendicular to the LOS vector. The equations for PE and BE are given in (4.3) and (4.4), respectively.

$$\dot{\theta}_E = K\left(\theta_E - \vartheta\right) \tag{4.3}$$

$$\dot{\theta}_E = K\text{sign}(\psi)\left(|\psi| - \frac{\pi}{2}\right), \psi = \pi + \vartheta - \theta_E \tag{4.4}$$

Implementation

Several important assumptions were made about the environment model to facilitate experimentation. First, it was assumed that each agent would have perfect information about the others, without any form of noise or uncertainty. This meant the input vector to the neural network was the true state vector. The second assumption was that all agents moved with constant speed. Lastly, the control signals were assumed to act without any form of inertia. This meant the agent would immediately experience the commanded latax and its heading would be updated accordingly. However, the Euler method was used to solve the kinematic equations (4.5) with a time step $\Delta t = 0.01$, so there was an effective delay between when the latax was commanded and when it would impact the trajectory of the agent. The potential for confounding interactions between the update rules of the environment model and the reward mechanism was mitigated by evaluating the behavior models only every five time steps of the environment. This allowed for sufficient change in the environment between evaluations to provide meaningful information to the training process while maintaining a

time resolution fine enough to ensure the capture phenomenon would not be missed.

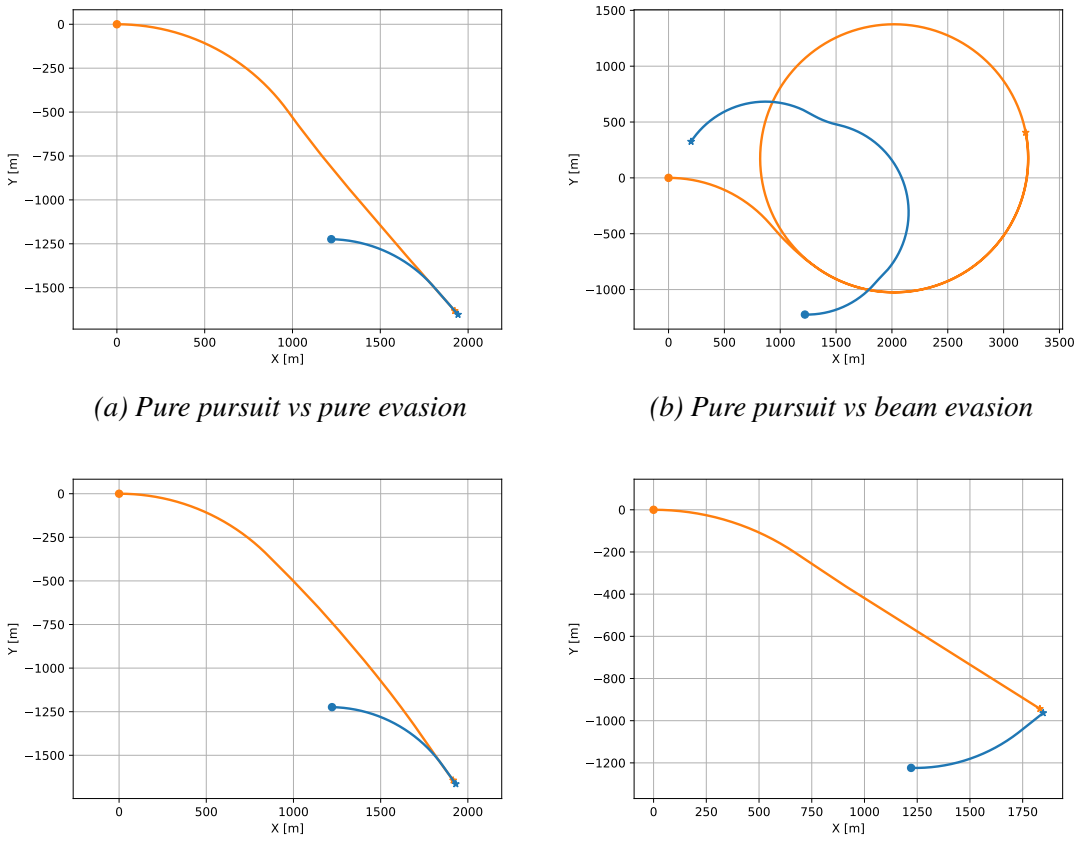
$$\begin{aligned} x(t+\Delta t) &= x(t) + v\cos\left(\theta(t)\right)\Delta t \\ y(t+\Delta t) &= y(t) + v\sin\left(\theta(t)\right)\Delta t \\ \theta(t+\Delta t) &= \theta(t) + \dot{\theta}(t)\Delta t \end{aligned} \tag{4.5}$$

The maximum turn rate $\dot{\theta}_{max}$ parameter was held constant at 0.50 radians per second for the pursuer and 0.25 radians per second for the evader. These values were based on notional turning capabilities of aircraft systems. The evader turn rate roughly equates to a $3g$ sustained turned, while the pursuer turns in excess of $10g$ [123].

The magnitude of the time step was based on the speed of the pursuer agent and the chosen capture radius $r_c = 30$ meters. The pursuers were given a speed of 600 meters per second, and the evaders 200 meters per second. The chosen time step of 10 milliseconds seconds resulted in a condition similar to the one-dimensional Courant-Friedrichs-Lewy condition (4.6) where the dimensionless Courant number C was less than 1 across all possible configurations. This meant the distance between the pursuer and evader could not change by more than the capture radius in a single time step, ensuring the environment would accurately track the capture condition.

$$\begin{aligned} C_p &= \frac{v_p\, dt}{r_c} = \frac{600m/s \times 0.01s}{30m} = 0.20 \\ C_{min} &= \frac{(v_p - v_e)dt}{r_c} = \frac{(600-200)m/s \times 0.01s}{30m} = 0.133 \\ C_{max} &= \frac{(v_p + v_e)dt}{r_c} = \frac{(600+200)m/s \times 0.01s}{30m} = 0.267 \end{aligned} \tag{4.6}$$

Each combination of pursuit and evasion algorithm was simulated on a single test geometry to ensure the models performed as expected, as well as to determine the computational cost associated with the models. The results shown in Figure 4.4 were obtained in a matter of seconds in terms of computational time. Several observations can be made: PE performs poorly, independent of pursuer guidance; PN performs well, independent of evader



(a) Pure pursuit vs pure evasion

(b) Pure pursuit vs beam evasion

(c) Proportional navigation vs pure evasion

(d) Proportional navigation vs beam evasion

Figure 4.4: Baseline pursuit-evasion trajectories. The pursuer is shown in orange and the evader in blue. The initial condition is the same for each case

guidance; and BE is effective against a pursuer using PP.

Designing for Pursuit-Evasion

It was necessary to establish how well the pursuit-evasion game satisfied criterion **1.b**. A small design problem was formulated around the game to determine this. The design attributes considered in this example were the speed and turn rate of the evader. The ranges for these design attributes are given in Table 4.1. The speed and turn rate of the pursuer were not varied.

A factorial design was implemented over the two-dimensional design space with 50 discretizations per dimension, yielding a total of 2,500 points. Each of these points was

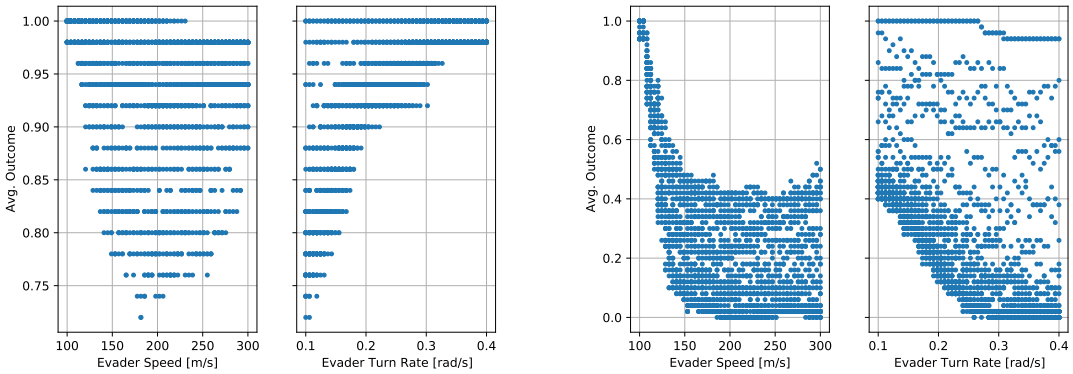
Table 4.1: Design attribute ranges for pursuit-evasion design problem

Attribute	Low Value	High Value	Unit
Speed	100	300	m/s
Turn rate	0.1	0.4	rad/s

simulated on a set of 50 geometries which were randomly generated, for a total of 125,000 simulations. The primary metric used here was the outcome, i.e. whether or not the evader was captured. This value was averaged over the 50 geometries to estimate the probability of intercept for each point in the design space. The results are shown in Figure 4.5.

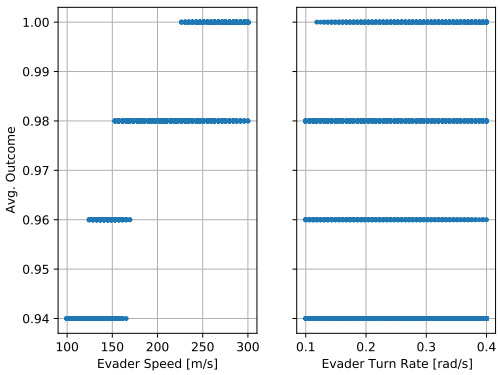
Several observations can be made about the results shown in Figure 4.5. First, the effect of design attributes on the effectiveness of either agent is evident, and especially so when the pursuer uses PP guidance. Second, in general, faster and more maneuverable evaders performed better. This agrees with the intuition developed by fighter pilots and documented in literature [123]. However, it can be seen that evaders using PE against a pursuer using PP performed best when they had low turn rates and middling speed, which is counterintuitive. This might be explained by the critical flaw in the PE guidance algorithm: An evader with a high turn rate will spend less time maneuvering into a tail chase, where it is practically guaranteed to lose. More time spent turning means more time with a non-zero LOS angle rate, approaching the effectiveness of the BE guidance algorithm.

BE guidance achieved perfect performance against PE in several cases, and more so when the evader was faster and more maneuverable. This makes sense, since the maneuverable evader could effectively cut across the path of the pursuer and force it to overshoot before turning back in a maneuver which might resemble a jink. However, BE was not significantly better than PE against pursuers using PN. This may be attributed to the optimality of PN as a pursuit guidance strategy [62]. However, the results show slower evaders performed slight better against PN compared to their faster counterparts. There was no clear reason for these phenomena. The only reasonable explanation was that the evader was able

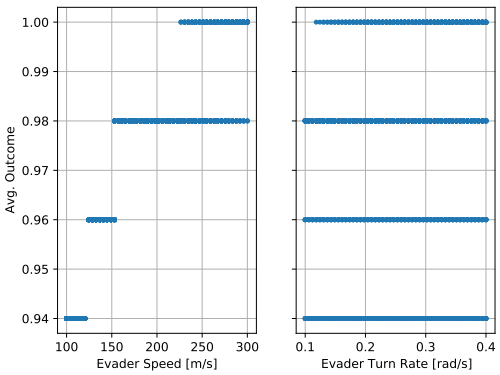


(a) PP vs PE

(b) PP vs BE



(c) PN vs PE



(d) PN vs BE

Figure 4.5: Capture rate as a function of design variable settings for evader design space exploration with baseline guidance algorithms

to avoid capture by maintaining a position inside the turn radius of the pursuer. Speed, turn radius, and turn rate are related by (4.7), which yields the insight that lower speed corresponds to tighter turns. The turn radius of the pursuer was $6000/0.5 = 1200$ meters, while that of the slowest evader with the lowest turn rate was $100/0.1 = 1000$ meters.

$$R_{turn} = \frac{u}{\omega} \tag{4.7}$$

Summary

The baseline analyses performed in this section establish the pursuit-evasion game as a suitable model for testing the proposed methodology. It satisfies all five criteria established at the outset, and the results produced here can be used to judge the fitness of the methodology on problems with these characteristics.

4.5.3 Defining Measures of Performance and Effectiveness

The agents were trained using a reward signal derived from the premise of the pursuit-evasion game and the available state information. In its simplest form, the objective of the pursuer was to quickly capture the evader. In other words, the pursuer sought to reduce the distance to the evader to the capture radius in as short a time as possible. The objective of the evader was exactly the opposite of this: To maintain the distance between itself and the pursuer above the capture radius for as long as possible. The running reward mechanisms (4.8) were designed to reflect these considerations. The reward was solely a function of the distance between the two agents, normalized between the capture radius $r_{cap} = 30m$ and escape radius $r_{esc} = 5000m$, and is zero-sum. It is worth noting that the "reward" to the pursuer is negative, and decreases with the normalized separation value. It is effectively a penalty which lessens as the pursuer gets closer to its main objective – capturing the evader. This choice of reward mechanism was intended to discourage a specific behavior: If the running reward were positive then the pursuer could increase its performance by

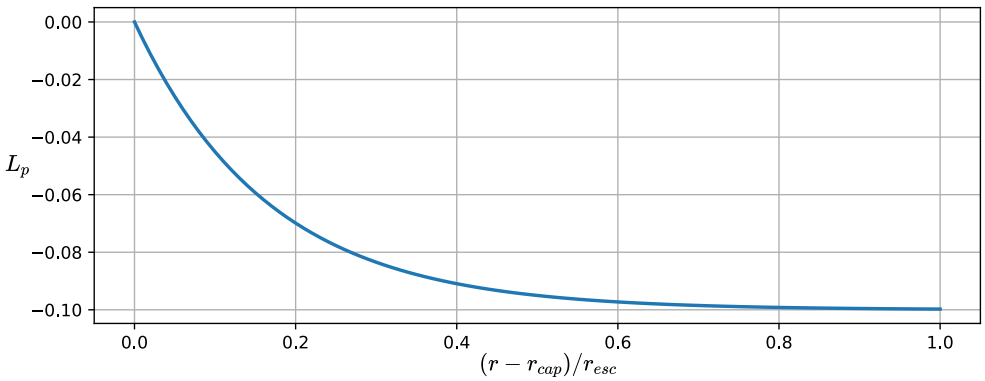


Figure 4.6: Pursuer reward versus normalized distance to the evader

delaying capture until the very end of the simulation. Conversely, with the reward being zero-sum, the evader is encouraged to delay capture as long as possible so as to accrue running rewards. The reward for the pursuer is visualized in Figure 4.6.

$$\begin{aligned} L_p &= 0.1\left(-1 + \exp\left(-6(r - r_{cap})/r_{esc}\right)\right) \\ L_e &= -L_p \end{aligned} \tag{4.8}$$

Terminal rewards (4.9) were implemented for the pursuer and evader, respectively, to further encourage the models towards their respective goals. The large terminal values, relative to the running rewards, were intended to reflect the importance of the terminal conditions. Furthermore, the escape condition was less penalizing to the pursuer because it was expected that the evader would not be able to escape except in the early stages of training. This was anticipated because the pursuer speed was three times that of the evader. The design of these mechanisms was not the focus of this experiment and those selected were deemed to adequately represent the desired property of higher rewards corresponding to more desirable states.

$$\begin{aligned} V_p &= \begin{cases} 100 & \text{if evader was captured} \\ -10 & \text{if evader escaped} \end{cases} \\ V_e &= -V_p \end{aligned} \tag{4.9}$$

4.5.4 Artificial Neural Network Architecture

The ANNs used for pursuit-evasion experiments were trained as deep stochastic policy networks using the PPO algorithm, which maps the state vector to scalar values corresponding to each available action. The action taken is determined by sampling a categorical distribution whose probability mass function is given by the softmax transformation (4.10) applied to the network output [13]. The state observations input to the neural networks are listed in Table 4.2. The subscript s denotes a state of the self, while the subscript o denotes a state of the opposing agent. These states were selected based on the work by Austin et al. [9].

$$P(a_i) = \frac{\exp(y_i)}{\sum_{j=1}^{n} \exp(y_j)} \tag{4.10}$$

Table 4.2: States for pursuit-evasion agent training

State	Equation	Symbol	Units
Range	$\sqrt{(x_s - x_o)^2 + (y_s - y_o)^2}$	r	kilometers
Relative Bearing	$\arccos\left(\frac{\vec{v}_s \cdot \vec{r}}{\|\vec{v}_s\|\|\vec{r}\|}\right)$	ω	radians
Relative Heading	$\arccos\left(\frac{\vec{v}_o \cdot \vec{r}}{\|\vec{v}_o\|\|\vec{r}\|}\right)$	θ	radians
Normalized Elapsed Time	$t_{elapsed}/t_{max}$	$\tilde{t}$	N.D.

The networks had two hidden layers, each with 50 nodes. The first layer used the hyperbolic tangent activation, while the second use the rectified linear unit. These choices were based on simple and limited preliminary experimentation, as well as prior experience. The network architecture is shown in Figure 4.7, where the output nodes use the linear activation function. The outputs of the model were used as the weights in a categorical distribution, which was sampled to perform action selection. An example algorithm for categorical distribution sampling was given in Algorithm . The selected action was mapped to the normalized commanded latax using (4.11) where x is the output from the ANN, which was subsequently mapped to a turn rate using (4.12).

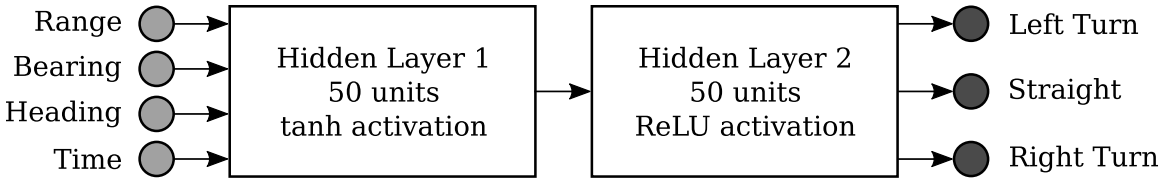


Figure 4.7: Network architecture for pursuit-evasion experiments

$$\lambda = -1 + \text{Categorical}(x) \in \{-1, 0, 1\} \tag{4.11}$$

$$\dot{\theta}(t) = \lambda \, \dot{\theta}_{max} \tag{4.12}$$

4.6 Experiment 1: Reinforcement Learning

The first experiment was intended to demonstrate the application of reinforcement learning to the problem of simulated aerial engagements as represented by a game of pursuit and evasion. This experiment was intended to provide evidence in support or refutation of Hypotheses 1 and 2 simultaneously. The hypotheses would be substantiated if the models produced by the application of RL could achieve MOEs and MOPs at least on par with the baselines. ANNs were trained to control each of the two agents in the pursuit-evasion scenario separately. The opposing, non-learning agent used each of appropriate baseline guidance algorithms. This resulted in four distinct cases:

1. Trained pursuer versus Pure evasion evader
2. Trained pursuer versus Beam evasion evader
3. Trained evader versus Pure pursuit pursuer
4. Trained evader versus Proportional navigation pursuer

4.6.1 Training Procedure

Twenty-four models were trained independently for each agent. The multi-agent training method was not employed because it was not needed. Each model was trained for 50,000 episodes. Each episode consisted of simulating engagements against an opponent using a randomly selected guidance algorithm from the baseline set.

The initial geometry of each simulation was randomly generated. The pursuer was always initialized at the origin with its heading along the inertial x-axis. The initial state of the evader was randomly sampled from the distributions (4.13). These distributions are shown as the shaded region in Figure 4.8. The relative bearing and heading of the evader were not restricted in an effort to develop more generalizable models. The lower and upper limits of the distribution over initial range – that is, separation between the agents – correspond to one-fifth and four-fifths of the escape radius, respectively.

$$r \sim \mathcal{U}\left(1000, 4000\right), \quad \omega \sim \mathcal{U}(-\pi, \pi), \quad \theta \sim \mathcal{U}(-\pi, \pi) \tag{4.13}$$

The number of simulations per episode was not prescribed. Instead, a minimum number of data samples collected for training was used to determine how many simulations would be performed. The number used for these experiments was 1,000 samples, and data collection was terminated after that threshold had been surpassed.

4.6.2 Testing the Models

The models were tested at regular intervals throughout the training process to gain better insights into how adaptation and learning progressed. A set of geometries, shown in Figure 4.9, was generated a priori and each model was tested on those geometries against both baseline opponent guidance algorithms. The geometries were sampled from the distributions (4.14). The test interval was 50 episodes, resulting in 1,000 test points for analysis. The performance of each model was tracked for each geometry using the total reward,

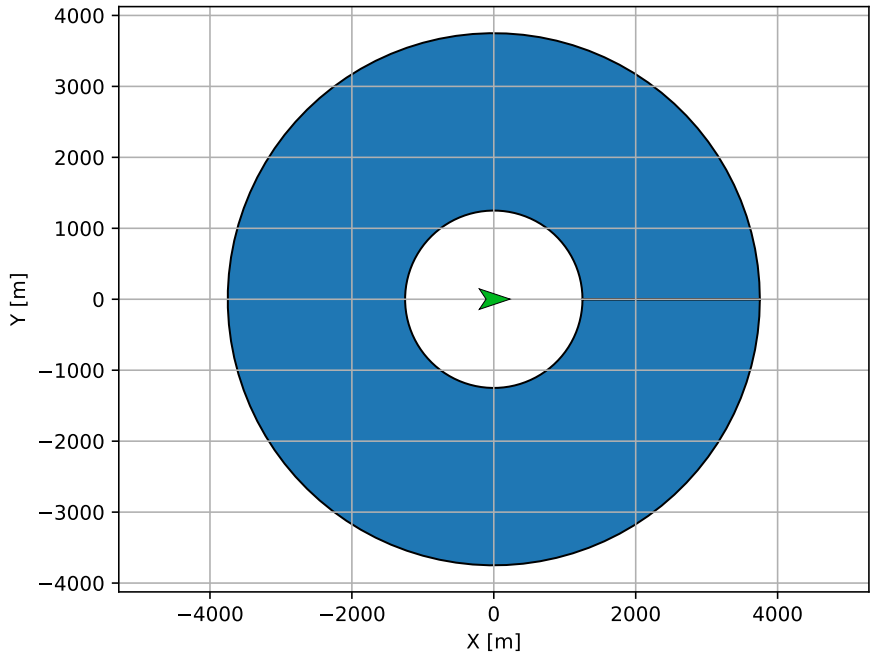


Figure 4.8: Valid initial positions for the evader during training. Pursuer, indicated by green wedge, always starts at the origin.

outcome, and end time as metrics.

$$r \sim \mathcal{U}(0.2\, r_{esc}, 0.6\, r_{esc}), \quad \omega \sim \mathcal{U}(-\pi/3, \pi/3), \quad \theta \sim \mathcal{U}(-\pi/3, \pi/3) \tag{4.14}$$

The metrics were recorded for simulations using each combination of the four baseline guidance algorithms. The results of the baseline simulations are given in Table 4.3. The Outcome column indicates which of two possible termination criteria was met first. An outcome of 1 indicates the evader was captured, while an outcome of 0 indicates either the evader escaped or 20 seconds of simulated time had elapsed without either an escape or capture occurring. Only the pursuer reward is provided because the engagement was modeled as a zero-sum game, meaning the evader reward was of equal magnitude and opposite in sign to the pursuer reward. These baseline results show that capture is possible in every test case, as well as that it would be possible for the evader to avoid capture in each

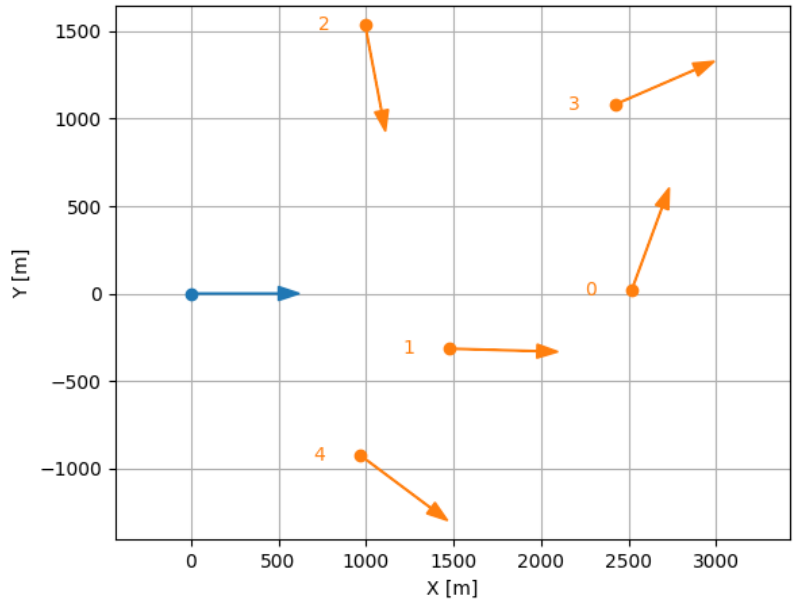


Figure 4.9: Initial conditions for pursuit-evasion test simulations

case against pure pursuit. However, the data also indicate potential difficult in countering the proportional navigation guidance algorithm.

Table 4.3: Baseline results for experiment 1

		Geometry					
	Case	**0**	**1**	**2**	**3**	**4**	*Average*
Pursuer Reward	PP vs PE	92.32	96.09	80.99	90.84	96.13	*91.27*
	PP vs BE	-27.60	-29.45	-27.96	-28.62	-28.76	*-28.48*
	PN vs PE	92.28	96.09	96.49	90.84	96.13	*94.37*
	PN vs BE	93.90	96.29	96.49	91.63	96.57	*94.98*
Outcome	PP vs PE	1	1	1	1	1	*1*
	PP vs BE	0	0	0	0	0	*0*
	PN vs PE	1	1	1	1	1	*1*
	PN vs BE	1	1	1	1	1	*1*
End Time [s]	PP vs PE	5.81	3.71	14.23	6.61	3.74	*6.82*
	PP vs BE	20	20	20	20	20	*20*
	PN vs PE	5.82	3.71	2.92	6.61	3.74	*4.56*
	PN vs BE	4.49	3.45	2.92	5.84	3.25	*3.99*

Statistical Analysis of Baselines

A set of 500 geometries was generated to allow statistical analysis of the models. The four combinations of baseline guidance algorithms were simulated on these 500 geometries in order to establish a datum. Histograms of the three metrics – reward, capture, and end time – are shown in Figures 4.10 and 4.11 for pursuers using PP and PN, respectively.

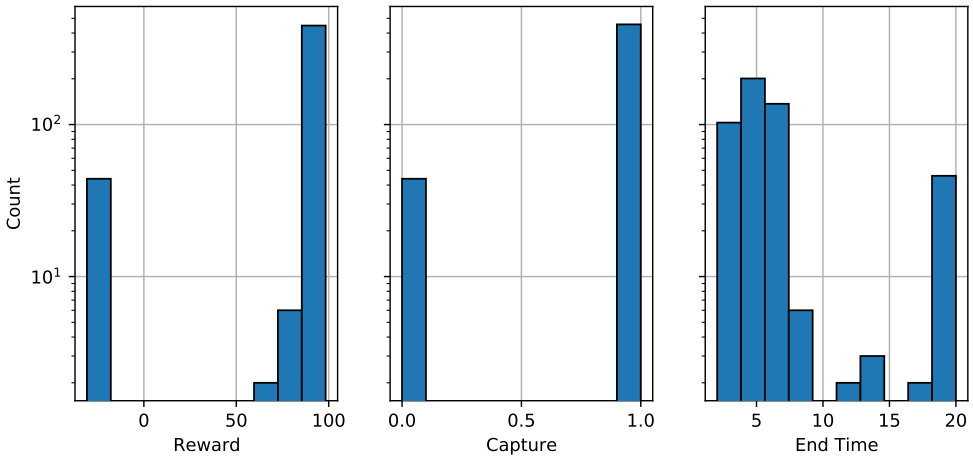
Figure 4.10 indicated the PP algorithm was fairly effective against PE but not so against BE, the latter having more failures than success. Bimodality in the reward metric is driven by the bivariate capture metric and its significant influence on the former: Failure to capture results in a terminal reward of -10, while a successful capture rewards +100. The remainder of the variability is driven by the end time metric. Failure to capture the evader was likely to result in an end time of 20 seconds. However, it was possible for the evader to escape in less than 20 seconds by crossing the 10 km separation threshold.

The data were divided into six groups among three categories: First, whether or not the evader was captured, and then by whether the simulation ended in more or less than 10 seconds. The cases where the evader escaped capture for more than 10 seconds were further split into cases where time expired, i.e. the end time was at least 20 seconds, and those where $10 < t_{end} < 20$. The latter corresponds to cases where the evader was able to achieve a separation from the pursuer greater than 10 km. The results are of these groupings are reported in Table 4.4.

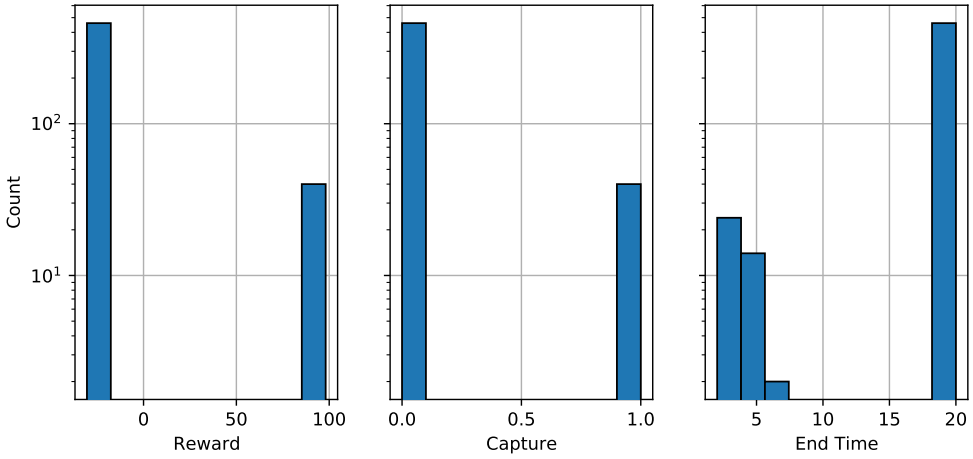
Table 4.4: Case count for each of four groups divided by capture and end time

Case	**Cap < 10s**	**Cap > 10s**	**Esc < 10s**	**Esc > 10s**	**Esc $\geq$ 20s**
PP vs PE	447	9	0	44	44
PP vs BE	40	0	0	460	460
PN vs PE	485	0	4	11	7
PN vs BE	480	0	0	20	7

Table 4.4 shows the majority of captures were achieved in less than 10 seconds. How-

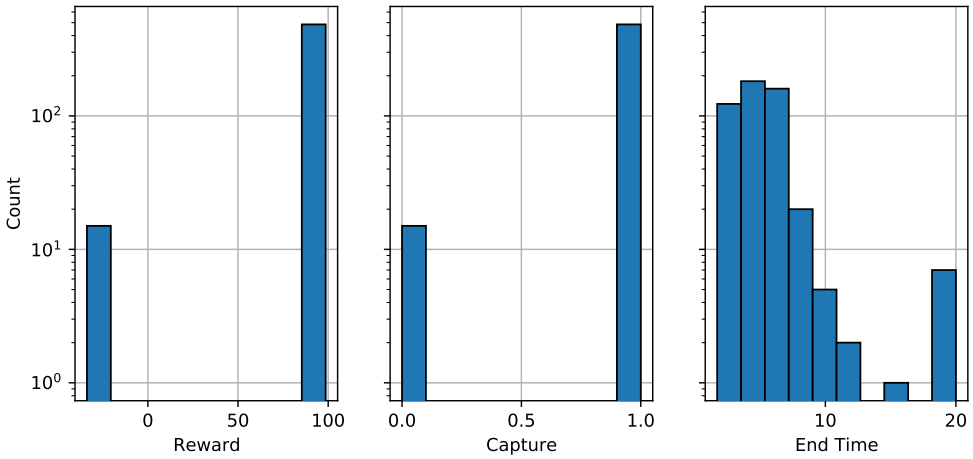


(a) Versus PE

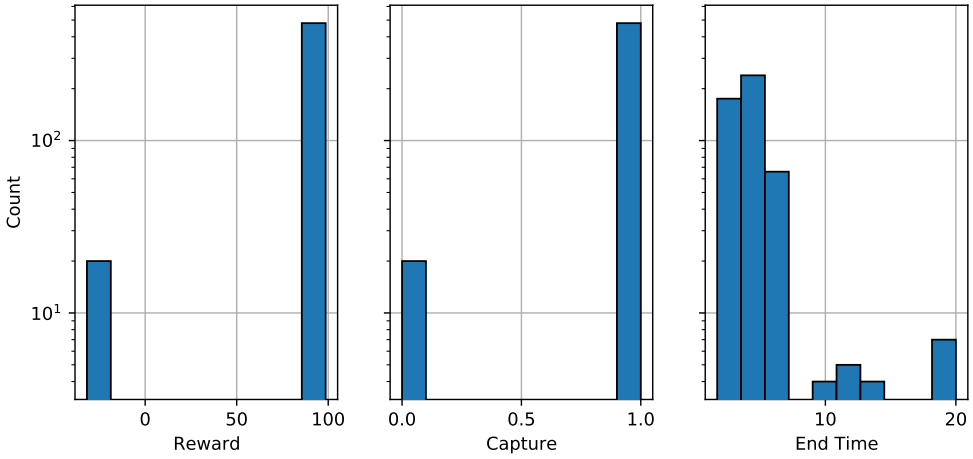


(b) Versus BE

Figure 4.10: Distributions of metrics for pursuers using PP



(a) Versus PE



(b) Versus BE

Figure 4.11: Distributions of metrics for pursuers using PN

ever, pursuers using PP were only able to capture evaders using BE in 40 of the 500 cases; the evader avoided capture for 20 seconds in the other 460 cases. However, pursuers using PE never allowed the evader to escape via the range threshold. By contrast, pursuers using PN allowed the evader to escape by exceeding the range threshold in 21 of 1,000 cases. The cause for this was attributed to a well-known flaw in the PN guidance algorithm: If the relative bearing to the target is greater than 60 degrees then the algorithm can struggle to turn the vehicle around and get back on track.

Selecting appropriate statistical measures to report was difficult because of the multimodal distributions observed. However, the groupings based on capture and end time provided useful divisions of the space for which localized statistics could be reported. Five groups were identified, corresponding to the columns in Table 4.4, listed below. The first two statistical moments were calculated for the reward and end time metrics. These data are reported in Table 4.5.

[Outcome 1] Capture in less than 10 seconds

[Outcome 2] Capture in more than 10 seconds

[Outcome 3] Range escape in less than 10 seconds

[Outcome 4] Range escape in more than 10 seconds

[Outcome 5] Time escape

4.6.3 Pursuer Results

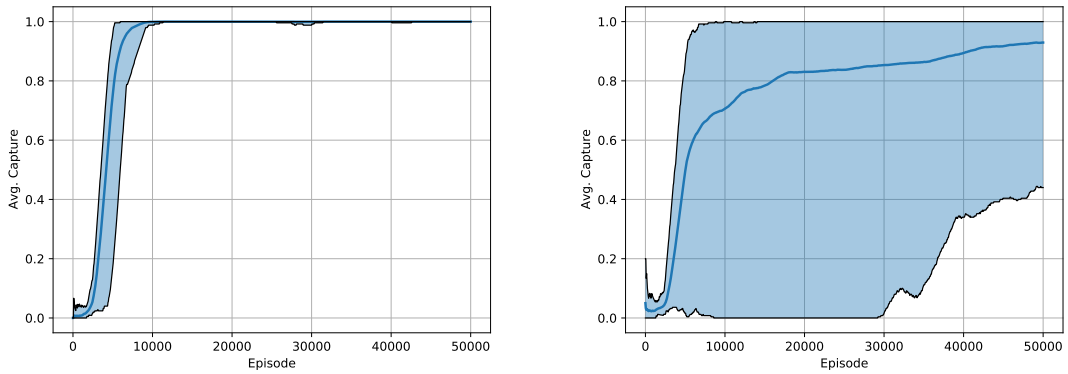
Performance versus Training Episode

The trends in performance, measured in terms of capture rate, for the pursuer models trained against evaders using pure evasion and beam evasion are shown in Figure 4.12. The plots show the average, minimum, and maximum moving average of captures across the five test geometries over the previous 50 test samples for each model. These metrics were calculated using (4.15), where i is the episode index and k is the model index. Each

Table 4.5: Performance statistics for baseline guidance algorithms on 500 test geometries grouped by capture and end time thresholds

Case	Outcome	Count	Reward		End Time	
PP vs PE	1	447	93.8333	2.4559	4.9271	1.3952
PP vs PE	2	9	78.3742	6.1482	15.3100	2.9304
PP vs PE	3	0	–	–	–	–
PP vs PE	4	0	–	–	–	–
PP vs PE	5	44	-28.9175	1.1532	20.0000	0.0000
PP vs BE	1	40	96.0826	1.6817	3.6137	1.0320
PP vs BE	2	0	–	–	–	–
PP vs BE	3	0	–	–	–	–
PP vs BE	4	0	–	–	–	–
PP vs BE	5	460	-28.1623	0.6035	20.0000	0.0000
PN vs PE	1	485	94.0648	2.5233	4.7523	1.4752
PN vs PE	2	0	–	–	–	–
PN vs PE	3	4	-25.2227	0.2889	9.8525	0.1153
PN vs PE	4	4	-27.2548	4.4090	12.2775	2.5396
PN vs PE	5	7	-30.5626	0.5801	20.0000	0.0000
PN vs BE	1	480	94.8348	2.0814	4.0344	1.1486
PN vs BE	2	0	–	–	–	–
PN vs BE	3	0	–	–	–	–
PN vs BE	4	13	-27.7613	1.8626	12.0808	1.5765
PN vs BE	5	7	-30.7641	0.6965	20.0000	0.0000

model performed poorly in the early episodes, which was expected. However, the models generally improved as training progressed, and did so rapidly within the first 10,000 episodes. There was little evidence of catastrophic forgetting, where the model performance rapidly degrades, but some models did not appear to improve significantly until very late in the training process, particularly against evaders using beam evasion. Several factors could have contributed to this, including random network initialization and evader



(a) Versus pure evasion *(b) Versus beam evasion*

Figure 4.12: Trends in test performance for ANN-controlled pursuers against evaders using baseline guidance algorithms

guidance randomization.

$$\begin{aligned} \bar{\bar{y}}_i &= \frac{1}{24}\sum_{k=0}^{23}\bar{y}_{i,k} \\ \lfloor \bar{y}_i \rfloor &= \min_k(\bar{y}_{i,k}) \\ \lceil \bar{y}_i \rceil &= \max_k(\bar{y}_{i,k}) \\ \bar{y}_{i,k} &= \frac{1}{50}\sum_{j=0}^{49} y_{i-j,k} \end{aligned} \tag{4.15}$$

Both figures show at least one model rapidly improving up to episode 10,000 and at least one model maintaining a perfect or near-perfect capture rate for the remainder of the episodes. The apparent variance in capture rate against pure evasion suggests the models had no trouble exploiting the simple guidance algorithm. However, high variance in capture rate was seen against beam evasion. Some models appeared capable of learning an effective strategy but at least one model was unable to achieve capture in a single test case until episode 30,000. The general trend in both cases was upward, suggesting additional training might see all 24 models achieve perfect performance in both test cases.

Final Model Performance on Test Geometries

The test data from the fully-trained pursuer agents against both evasion algorithms are provided in Table 4.6. The average performance over the 24 individual models is reported, along with the metrics of the model with the best average performance in each category. These metrics were selected based on the prior discussion regarding uncertainty in model initialization and training.

All of the models were able to achieve perfect capture rate against an evader using pure evasion. The capture rate against beam evasion was also quite high, although not perfect. This reflects the high variance seen in Figure 4.12b. However, the distribution is skewed towards the higher-performing end of the spectrum, suggesting a small number of outliers at the low end.

Table 4.6: Trained pursuer metrics against baseline evader

		Geometry					
	Case	**0**	**1**	**2**	**3**	**4**	*Average*
Pursuer Reward	Avg. vs PE	92.16	96.03	91.96	90.44	96.13	*93.35*
	Best vs PE	92.33	96.04	96.49	90.80	96.13	*94.36*
	Avg. vs BE	78.84	85.86	88.66	86.40	86.16	*85.18*
	Best vs BE	94.10	96.28	96.49	91.64	96.58	*95.02*
Outcome	Avg. vs PE	1	1	1	1	1	*1*
	Best vs PE	1	1	1	1	1	*1*
	Avg. vs BE	0.88	0.92	0.96	0.96	0.92	*0.93*
	Best vs BE	1	1	1	1	1	*1*
End Time [s]	Avg. vs PE	5.90	3.76	5.90	6.84	3.74	*5.23*
	Best vs PE	5.80	3.75	2.92	6.63	3.74	*4.57*
	Avg. vs BE	6.34	4.87	5.32	6.52	4.66	*5.54*
	Best vs BE	4.35	3.45	2.92	5.84	3.25	*3.96*

The data in Table 4.6 show the best ANNs performed about as well as the proportional navigation guidance algorithm in each of the five test geometries. Capture was achieved

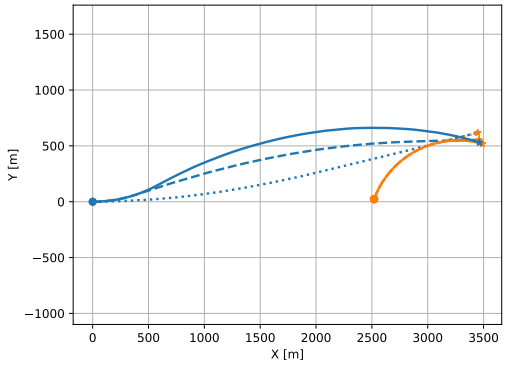
in each geometry; the highest average rewards against both pure evasion and beam evasion were within 1% of the proportional navigation rewards; and the average end times were almost identical. Together, these observations suggest the trained models were able to discover highly effective guidance models from scratch. More training with optimized hyperparameters may be able to push performance even further.

Visual Comparison of Test Trajectories

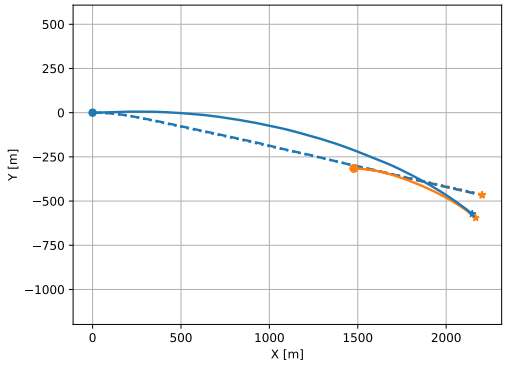
Figure 4.13 and Figure 4.14 show the trajectory generated by simulating the pursuer model with the highest average reward against evaders using pure evasion and beam evasion, respectively. Trajectories generated by pursuers using pure pursuit and proportional navigation guidance algorithms are shown as dotted and dashed lines, respectively.

Statistical Testing

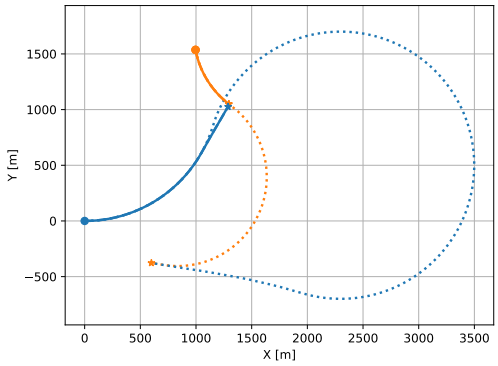
A single model was selected from the 24 for further statistical analysis. A selection was made using TOPSIS. Each of the fully trained models was tested on 500 pre-generated geometries and against both evader guidance algorithms in order to generate criteria for selection. Three separate analyses with TOPSIS were run, each with a different set of criteria. The first used the boolean capture metric, the second used the end time metric, and the third used the reward metric which included considerations for both capture and end time. The geometries used for this test were the same as those used to fill the cells of Table 4.5 to allow for a fair comparison against the baseline data. The top three results for each set of criteria are reported in Table 4.7. The similarity metric s_b was calculated using (4.16), where a value closer to 0 indicates closer proximity to the ideal solution. The results were inconsistent – that is, the choice model depended on the choice of metric. Closer inspection of the results showed Pursuer 0 ranked 14^{th} in the End Time metric with a similarity of 0.0962. A final round of TOPSIS was run with both the capture and end time metrics being considered simultaneously. The result is given in the last row of Table 4.7,



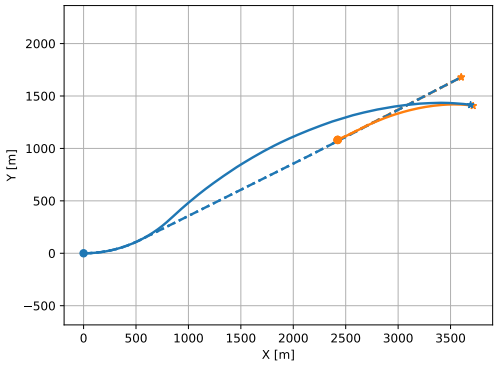
(a) Geometry 0



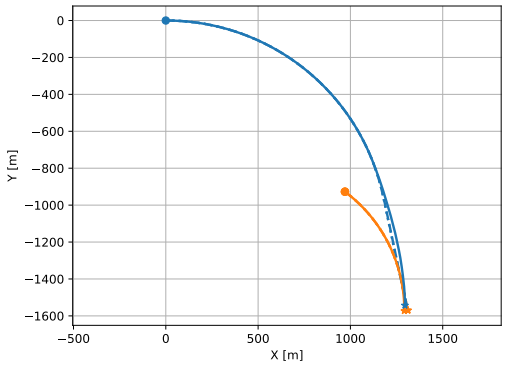
(b) Geometry 1



(c) Geometry 2

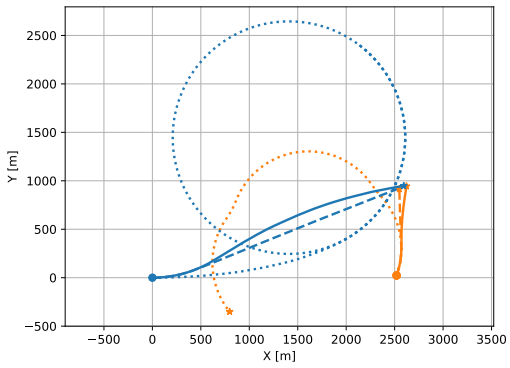


(d) Geometry 3

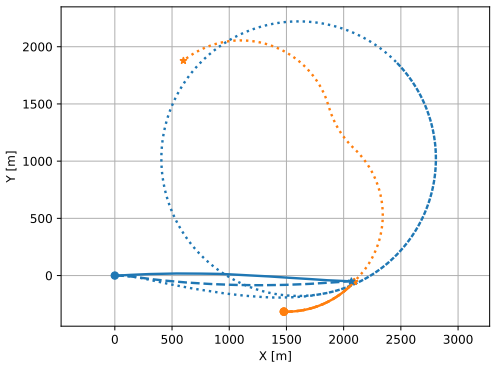


(e) Geometry 4

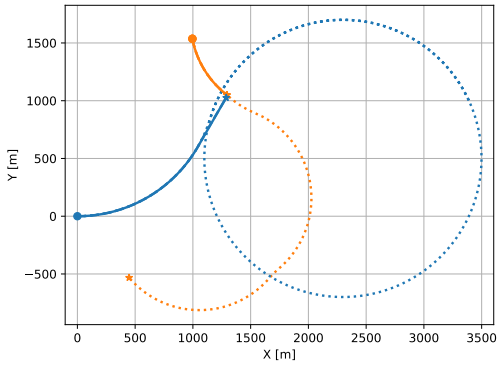
Figure 4.13: Trained pursuer model trajectories versus evaders using pure evasion. Trajectories from baseline proportional navigation guidance are shown as dashed lines.



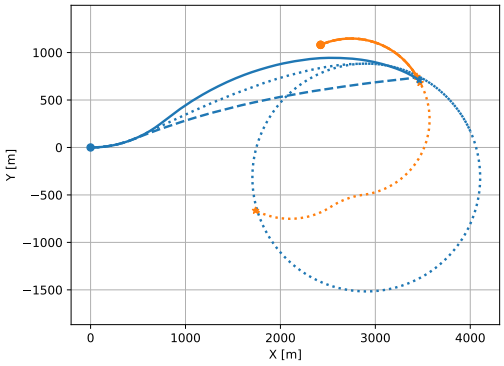
(a) Geometry 0



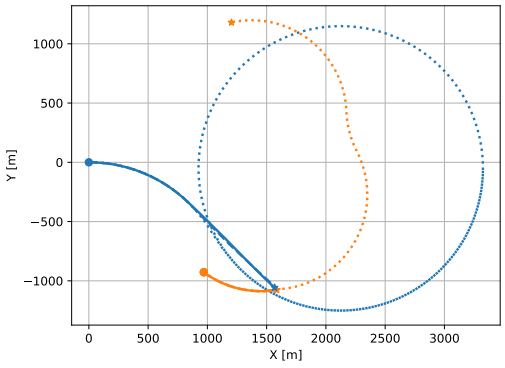
(b) Geometry 1



(c) Geometry 2



(d) Geometry 3



(e) Geometry 4

Figure 4.14: Trained pursuer model trajectories versus evaders using beam evasion. Trajectories where the pursuer used proportional navigation and pure pursuit guidance are shown as dashed and dotted lines, respectively.

showing the pursuer with index 0 performed best overall. This outcome might have been guessed from the rankings with capture and reward as the criteria, but it was worthwhile to confirm the choice of model to carry forward.

$$s_b = \frac{d_{best}}{d_{best} + d_{worst}} \tag{4.16}$$

Table 4.7: Results from applying TOPSIS to pursuers using multiple sets of criteria

Criteria	**Rank 1**		**Rank 2**		**Rank 3**	
	Index	s_b	Index	s_b	Index	s_b
Capture	0	0.0788	1	0.3197	21	0.3197
End Time	21	0.0144	1	0.0147	9	0.0150
Reward	0	0.0867	9	0.1905	21	0.1905
Final	*0*	*0.0933*	*21*	*0.1544*	*1*	*0.1545*

Histograms of reward, capture, and end time for all simulations against evaders using PE are shown in Figure 4.15, and those against evaders using BE are shown in Figure 4.16. The reward data is bimodal because it is strongly influenced by whether or not capture was achieved, which is necessarily binomial. End time data was also bimodal, with one cluster at the low end and another at the high end. An end time of 20 seconds would indicate failure to capture the evader.

The first metric of interest was the probability of capturing the evader, independent of initial condition. The rate at which each pursuer captured evaders was estimated by bootstrapping 5,000 samples of size 100 from the 500 data points for each pursuer model against each evader model. Distributions of the bootstrapped data for each pursuer against evaders using PE and BE are shown in Figures 4.17 and 4.18, respectively. The Central Limit Theorem was then applied in order to get an unbiased estimate of the true rate parameter. Further, a two-sample t test was performed on each pair of pursuer models against each evader model to test for significance. The results, given in Table 4.8, show the cap-

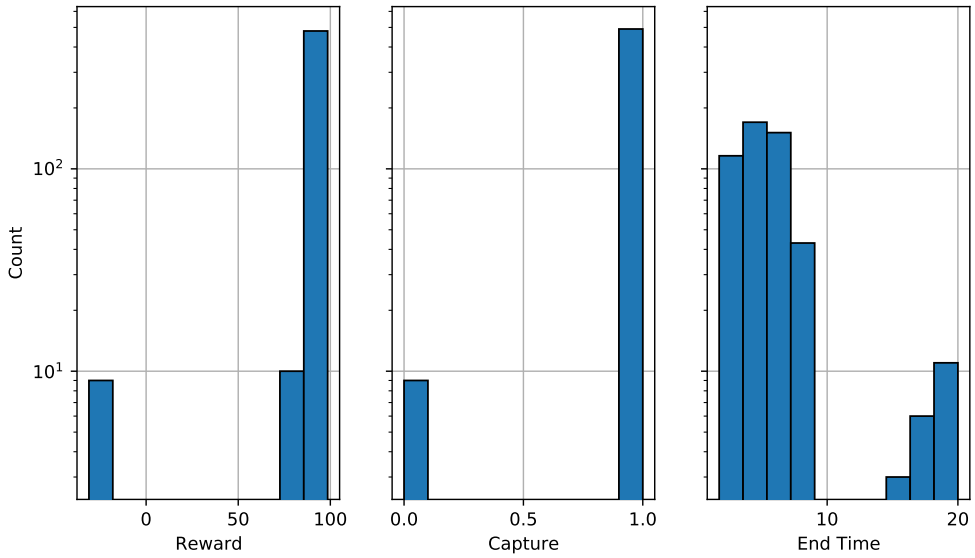


Figure 4.15: Histograms of best pursuer metrics against evader using PE

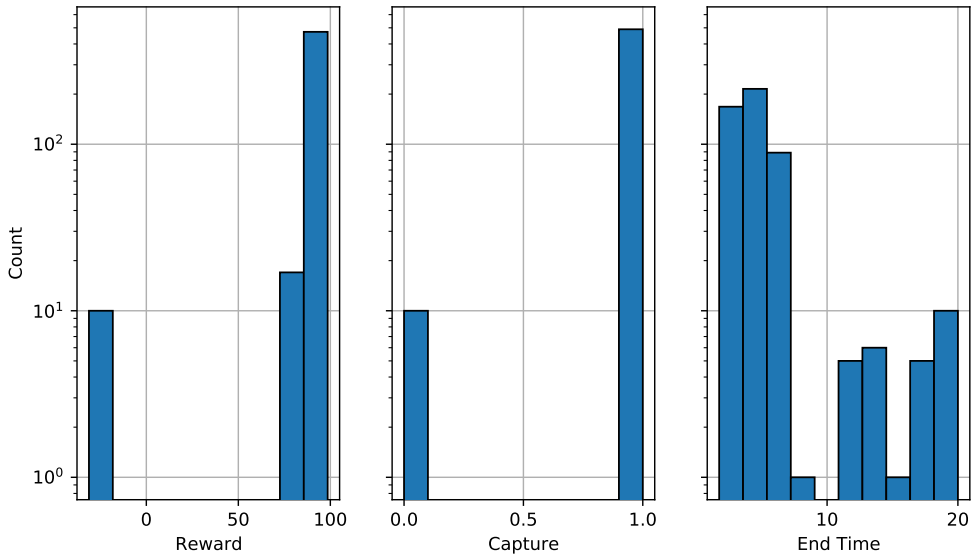


Figure 4.16: Histograms of best pursuer metrics against evader using BE

ture rate of the ANN-controlled pursuer was significantly better than either baseline against both evaders independent of initial condition.

Table 4.8: Results of capture rate testing for ANN-controlled pursuers

Baseline	**Evader**	$\bar{x}_0$	s_0	$\bar{x}_1$	s_1	t	ν
PP	PE	0.912	0.0282	0.982	0.0133	159.3	7078
PN	PE	0.970	0.0171	0.982	0.0133	41.11	9412
PP	BE	0.081	0.0270	0.980	0.0138	2099	7453
PN	BE	0.960	0.0195	0.980	0.0138	58.67	9002

Next, a pairwise comparison of capture was conducted between the ANN and baseline pursuit algorithms on each geometry. There were four possible outcomes of each pairwise test: (1) Both captured the evader, (2) both failed to capture the evader, (3) the ANN captured but the baseline did not, or (4) the baseline captured but the ANN did not. The first two were considered neutral outcomes, while the latter two presented an important test of effectiveness. If the ANN could capture the evader in more cases than the baselines then it could be said to be more effective. The results of the pairwise tests are given in Table 4.9. Outcome 4 was only realized in three times over 2,000 simulations, and the pursuer was using PN in all three cases. On the other hand, Outcome 3 was realized a total of 504 times with the ANN always being able to capture the evader in at least 6 cases where a baseline guidance algorithm could not. These results indicated the ANN-controlled pursuer was more effective than either baseline guidance algorithm when controlling for the initial condition of the engagement.

The data on simulation end time where the evader was captured were separated into two cases: Those where the simulation ended in less than 10 seconds, and those where it ended after more than 10 seconds. The resulting distributions are shown in Figures 4.19 and 4.20 for evaders using PE and BE, respectively. The cases where the simulation ended in less than 10 seconds appear close to normally distributed, while those greater than 10 seconds show strong negative skew. This may be explained by a simple phenomenon:

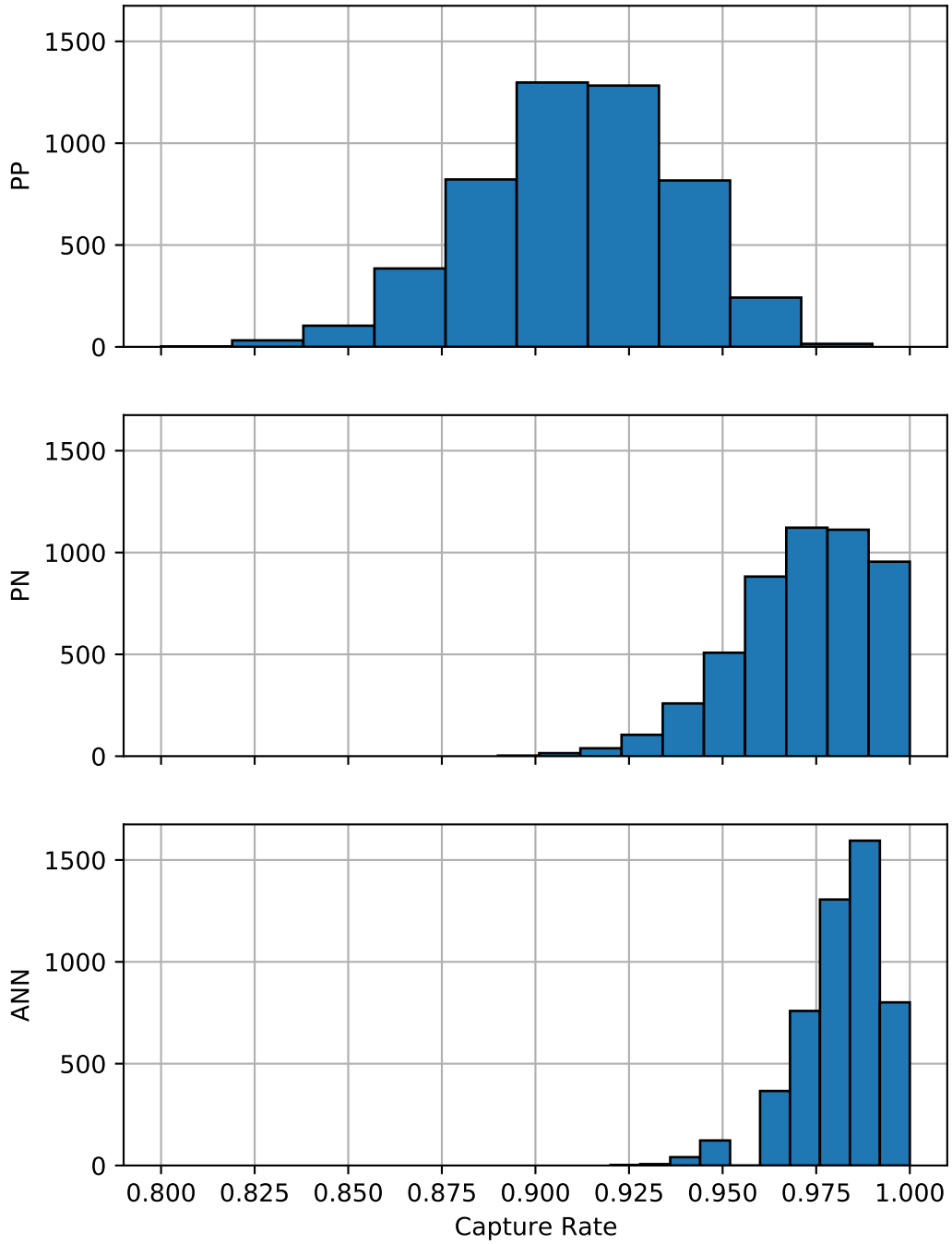


Figure 4.17: Distribution of estimated capture rates against evaders using PE

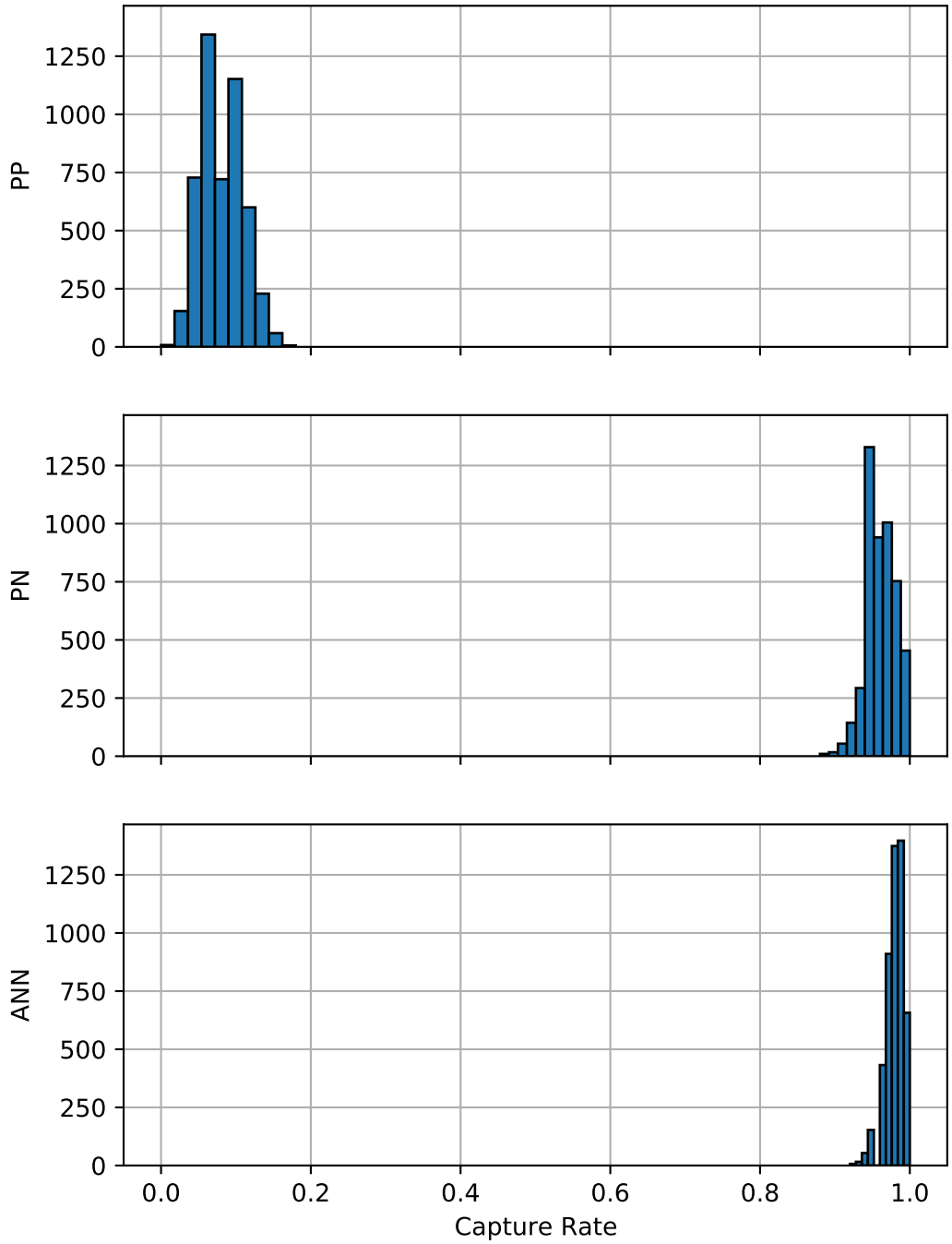
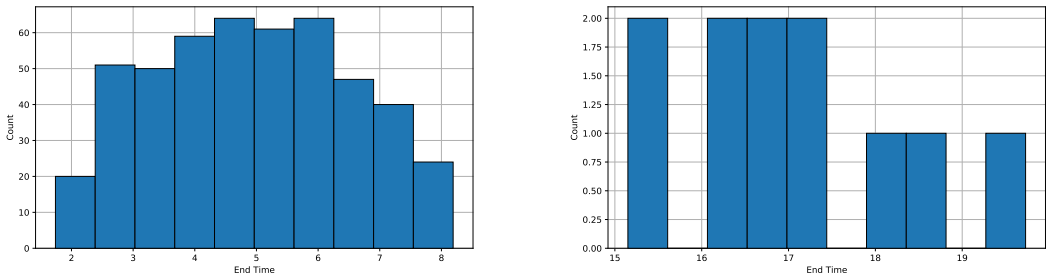


Figure 4.18: Distribution of estimated capture rates against evaders using BE

Table 4.9: Results of pairwise capture test between ANN and baselines

Baseline	Evader	Outcome 1	Outcome 2	Outcome 3	Outcome 4
PP	PE	456	9	35	0
PP	BE	40	10	450	0
PN	PE	485	9	6	0
PN	BE	477	7	13	3
Total		*1458*	*35*	*504*	*3*



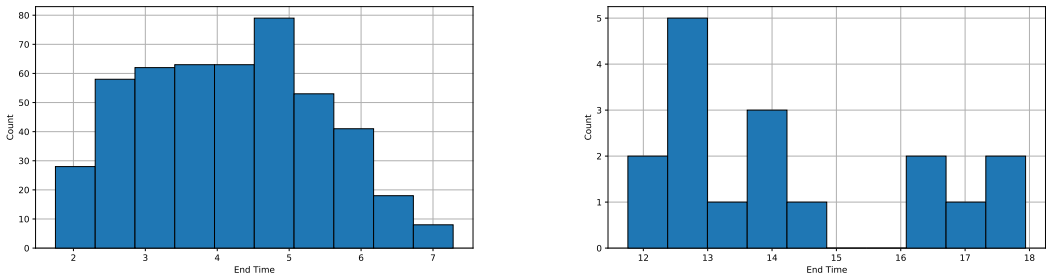
(a) Capture in less than 10s *(b) Capture in more than 10s*

Figure 4.19: Distributions of end time when capturing evader using PE

If the pursuer misses an early opportunity to capture the evader then it has to execute a wide turn to get back on track and line up for another opportunity. The turn radius of the pursuer can be calculated from its speed and turn rate: $r_{turn} = (600m/s)/(0.5rad/s) = 1200m$. The circumference of the turn can then be calculated: $c_{turn} = 2\pi r_{turn} \approx 7539.8m$. Finally, the time required to execute the turn at constant speed can be calculated: $t_{turn} = (7539.8m)/(600m/s) \approx 12.57s$. This provides a rough estimate of the time required to capture the evader given an initial miss, the actual value of which would depend on how the evader maneuvered over the course of the turn.

The data generated by the best-performing pursuer model on the 500 test geometries was grouped according to the capture and end time criteria without regard for the model which generated it. Summary statistics of the data for each evader guidance algorithm were then calculated, the results of which are reported in Table 4.10.

A series of statistical tests were then conducted on the available data. Specifically, a t



(a) Capture in less than 10s *(b) Capture in more than 10s*

Figure 4.20: Distributions of end time when capturing evader using BE

Table 4.10: Performance statistics for the best pursuer on 500 test geometries grouped by capture and end time thresholds

Case	Outcome	Count	Reward		End Time	
			$\bar{x}$	s	$\bar{x}$	s
vs PE	1	480	93.7939	2.7341	4.9284	1.5880
vs PE	2	11	75.6844	2.3313	16.9973	1.3559
vs PE	3	0	–	–	–	–
vs PE	4	0	–	–	–	–
vs PE	5	9	-29.1428	1.7965	20.0000	0.0000
vs BE	1	473	94.6395	2.2373	4.1773	1.2559
vs BE	2	17	80.9203	3.5799	14.2000	2.0992
vs BE	3	0	–	–	–	–
vs BE	4	0	–	–	–	–
vs BE	5	10	-29.4210	1.6511	20.0000	0.0000

test was used to compare the reward and end time metrics between the ANNs and baseline guidance algorithms for each group and against each evader algorithm. The null hypothesis for each test was: H_0*: The expected performance of the ANN and baseline guidance are equal.* Statistical testing could not be performed for cases where either group was empty, and this was the case for 11 of the 20 possible pairs. The results of the nine pairs where testing was possible are reported in Table 4.11.

Most of the cases have very high degrees of freedom, which was driven by the large

sample size for the ANN models. The last column in Table 4.11 indicates the lowest two-tailed significance level at which the corresponding null hypothesis could be rejected. The test results are to be interpreted in different ways for the reward and end time metrics. In either case, if p is high then it would not be reasonable to reject the null hypothesis that the difference in performance between the two models was statistically significant. If, however, p is low then it would be reasonable to reject the null hypothesis. For significant results, a test statistic $t > 0$ on the reward indicates the ANN out-performed the baseline and $t < 0$ indicates the ANN performed worse. A test statistic $t < 0$ on the end time indicates the ANN out-performed the baseline and $t > 0$ indicates the ANN performed worse. The comparisons between end times for Group 5 were not performed because they were meaningless; however, testing the reward could indicate how well the pursuers were able to track the evader before time had expired.

Table 4.11: Results of statistical testing between pursuer groups

Baseline	**Evader**	**Outcome**	**Metric**	t	ν	p
PP	PE	1	Reward	-0.2308	1.99e+08	0.819
PP	PE	1	End Time	0.0130	1.99e+08	0.99
PP	PE	2	Reward	-1.5702	843	0.116
PP	PE	2	End Time	1.9289	939	0.054
PP	PE	5	Reward	-0.5019	1365	0.617
PP	BE	1	Reward	-3.9998	4.92e+05	<0.001
PP	BE	1	End Time	2.7675	3.94e+05	0.006
PP	BE	5	Reward	-6.3367	1186	<0.001
PN	PE	1	Reward	-1.6027	2.22e+08	0.110
PN	PE	1	End Time	1.7881	2.22e+08	0.074
PN	PE	5	Reward	2.3515	768	0.018
PN	BE	1	Reward	-1.3978	2.14e+08	0.164
PN	BE	1	End Time	1.8366	2.13e+08	0.067
PN	BE	5	Reward	2.2755	1131	0.023

The conclusions that would be drawn from Table 4.11 were as follows:

- For Outcome 1 against PE, the ANNs performed slightly worse than PP and slightly worse than PN in both reward and end time, but the results were not sufficiently different to warrant rejecting the null hypothesis
- For Outcome 1 against BE, the ANNs performed worse than PP by a significant margin and worse than PN by moderately significant margin
- For Outcome 2 against PE, the ANNs performed worse than PP by a moderate margin

Summary

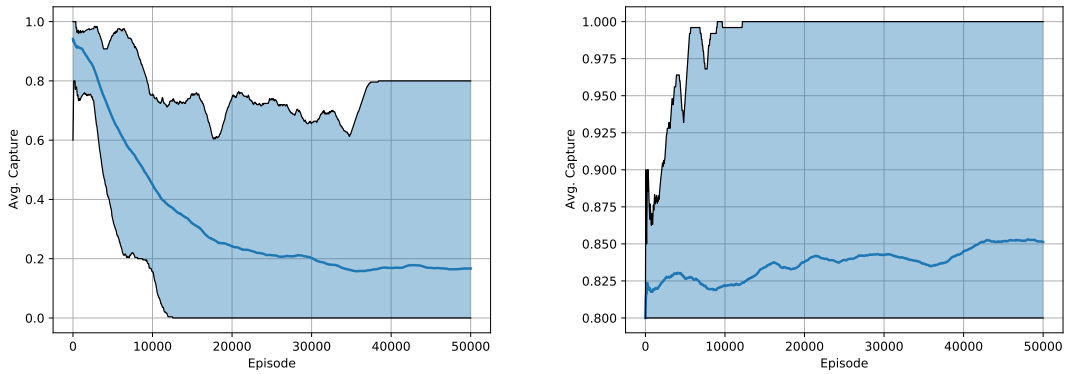
The results of the various statistical tests presented in this section indicated an application of RL with ANNs could be used to produce models of behavior which achieve levels effectiveness comparable to and even exceeding existing models of behavior. This came at the cost of slight degradation in performance compared to the baseline models. More training might allow the ANNs to close this gap.

TOPSIS was applied one more time to all the pursuer models, including the baselines, using both capture and end time against both PE and BE as the criteria. Pursuer 0 maintained its position as the top-ranked alternative with a similarity of 0.0991; Pursuer 21 also maintained its second-place position with a similarity of 0.1738; PN ranked 13th with a similarity of 0.1848; and PP ranked last with a similarity of 0.7668.

4.6.4 Evader Results

Performance versus Training Episode

The trends in capture rate for the trained evader models against pursuers using pure pursuit and proportional navigation are shown in Figure 4.21. The trend lines are the moving average over the previous 50 test points, each averaged over the five test geometries. It was apparent that the evaders were having success in avoiding capture against pure pursuit and were steadily improving to that end. However, avoiding capture against proportional navi-



(a) Versus pure pursuit *(b) Versus proportional navigation*

Figure 4.21: Trends in test performance for ANN-controlled evaders against pursuers using baseline guidance algorithms

gation was apparently much more difficult, as indicated by the y-axis scale in Figure 4.21b. Furthermore, at least one model saw significant degradation in performance and was unable to avoid capture in any of the test simulations for the majority of the training process, as indicated by the upper limit in Figure 4.21b.

Final Model Performance on Test Geometries

The five-case test data for the fully-trained evader models are presented in Table 4.12. The data show how the models performed very well against pure pursuit, with a low capture rate and high end times. The differences between the. The differences between the average and best metrics against proportional navigation were very small in the majority of cases, the lone exception being the geometry 2, where the models appeared to have discovered and exploited a flaw in the pursuit guidance. This was not entirely unexpected because of the known effectiveness of proportional navigation as a pursuit guidance algorithm.

Visual Comparison of Test Trajectories

Trajectories were generated using the fully-trained evader model with the highest average reward, simulated against pursuers using pure pursuit and proportional navigation. The re-

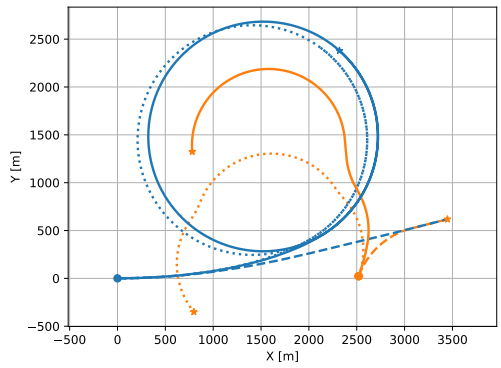
Table 4.12: Trained evaders metrics against baseline pursuers. Pursuer reward is reported for consistency, but the models were trained to minimize this metric.

		Geometry					
	Case	**0**	**1**	**2**	**3**	**4**	*Average*
Pursuer Reward	Avg. vs PP	-30.02	1.84	-14.78	-4.88	-3.38	*-10.25*
	Best vs PP	-30.05	-29.47	-29.80	-30.47	-29.53	*-29.87*
	Avg. vs PN	93.85	96.25	4.40	91.44	96.50	*76.49*
	Best vs PN	93.26	96.12	-26.37	91.63	96.31	*70.19*
Outcome	Avg. vs PP	0	0.25	0.12	0.21	0.21	*0.16*
	Best vs PP	0	0	0	0	0	*0*
	Avg. vs PN	1	1	0.25	1	1	*0.85*
	Best vs PN	1	1	0	1	1	*0.80*
End Time [s]	Avg. vs PP	20.00	15.91	18.87	17.19	16.59	*17.71*
	Best vs PP	20.00	20.00	20.00	20.00	20.00	*20.00*
	Avg. vs PN	4.46	3.50	8.46	6.02	3.33	*5.16*
	Best vs PN	4.99	3.69	10.35	5.84	3.56	*5.69*

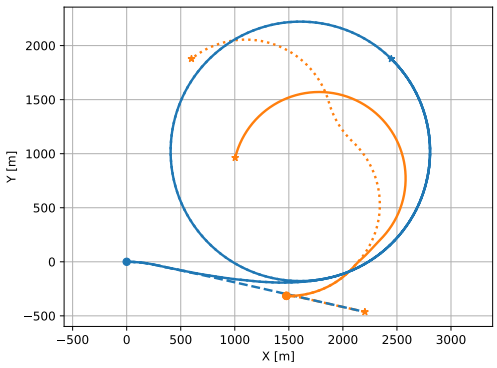
sults are shown in Figure 4.22 for pure pursuit and Figure 4.23 for proportional navigation. The paths traced by the ANN-controlled evader are shown as solid lines, while those traced by evaders using pure evasion and beam evasion are shown as dashed and dotted lines, respectively.

The trajectories generated by the ANN-controlled evader were visually similar to those generated by the beam evasion algorithm. This indicated the evader had learned some form of control which approximates the beam evasion algorithm, which was shown to be highly effective against pursuers using pure pursuit. It also appeared as though the evader had learned to maintain its position inside the turn radius of the pursuer using pure pursuit. This would have made capture practically impossible for the simple guidance algorithm.

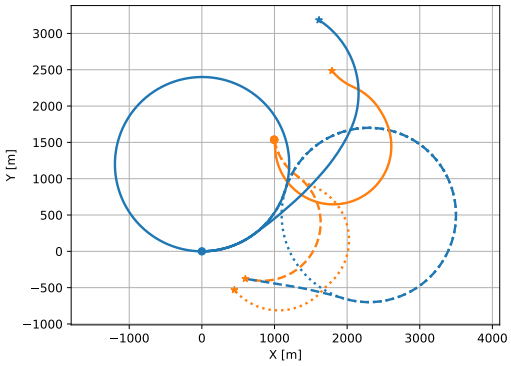
The trajectories against proportional navigation provided fewer insights, largely because the evader was unable to avoid capture in the majority of cases. However, it could again be said that the trajectories generated by the ANN-controlled evader were similar to



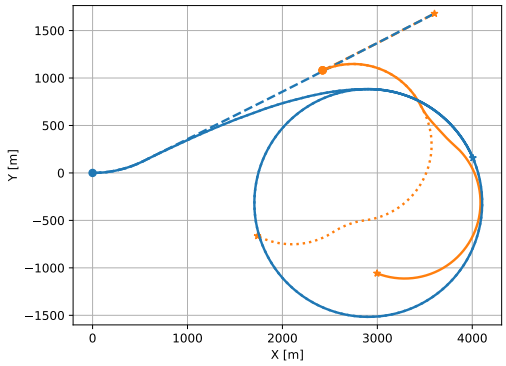
(a) Geometry 0



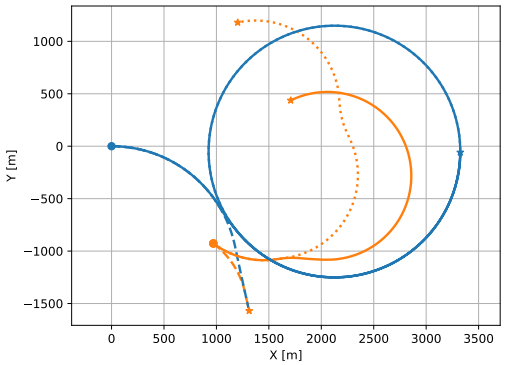
(b) Geometry 1



(c) Geometry 2



(d) Geometry 3



(e) Geometry 4

Figure 4.22: Trained evader model trajectories versus pursuers using pure pursuit. Dashed and dotted lines are trajectories for evaders using pure and beam evasion. respectively.

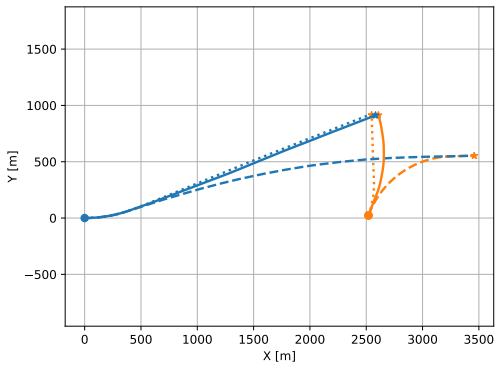
those generated by the beam evasion algorithm. The anomaly was geometry 2, where the ANN-controlled evader was able to avoid the initial capture opportunity seen in both the beam and pure evasion cases. It appeared to have done so by taking a shallower turn to stay within the turn radius of the pursuer. This seemed to have caused the proportional navigation algorithm to fail, as the pursuer did not complete its turn and instead continued off away from the evader. This may be an example of ANNs trained using RL techniques finding and exploiting errors in the models they are trained on. It is worth reiterating that the action selection is a stochastic process, and several attempts to replicate this trajectory resulted in the evader being captured near the point where the baseline algorithms were captured. Also, the ANNs could select a zero turn rate action which was not available to the beam or pure evasion algorithms.

Statistical Testing

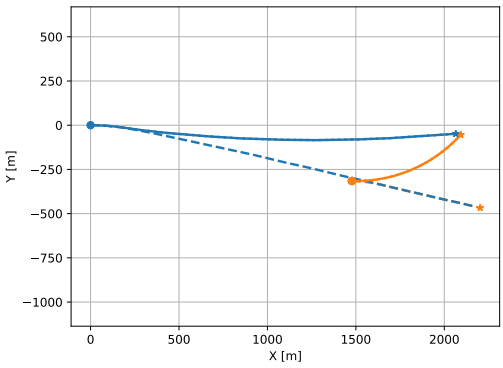
Each of the evader models was tested on 500 geometries to allow for the same statistical analyses which were conducted on the pursuer models. The process of applying TOPSIS to each metric individually and all three together was repeated, the results of which are given in Table 4.13. Rankings for the evaders were slightly more consistent than for the pursuers; the capture and reward metrics yielded identical rankings at the top. However, the rankings for end time were significantly different than the other two. When using the End Time metric in TOPSIS, Evader 7 ranked 6^{th} with a similarity metric of 0.4212. When using the Capture metric, Evader 22 ranked 15^{th} with a similarity metric of 0.1183.

Histograms of reward, capture, and end time for all simulations against pursuers using PP are shown in Figure 4.24, and those against pursuers using PN are shown in Figure 4.25. The evader was never captured in more than 10 seconds.

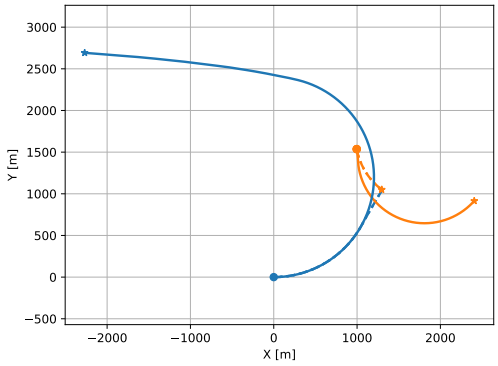
Bootstrap sampling of capture rate was performed for each combination of evader guidance against the baseline pursuer guidance algorithms for two-sample t testing. The same procedure was used here as was for the pursuers, and the results are presented in Table 4.14



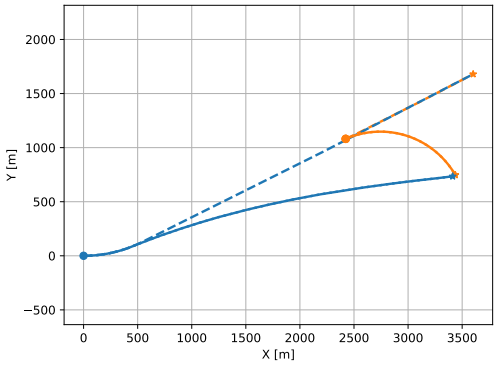
(a) Geometry 0



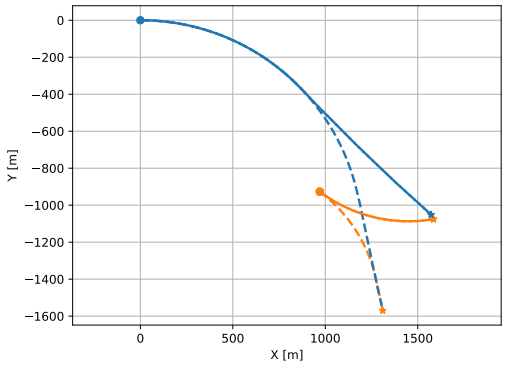
(b) Geometry 1



(c) Geometry 2



(d) Geometry 3



(e) Geometry 4

Figure 4.23: Trajectories from simulation of evader models trained against proportional navigation. Dashed and dotted lines are trajectories for evaders using pure and beam evasion. respectively.

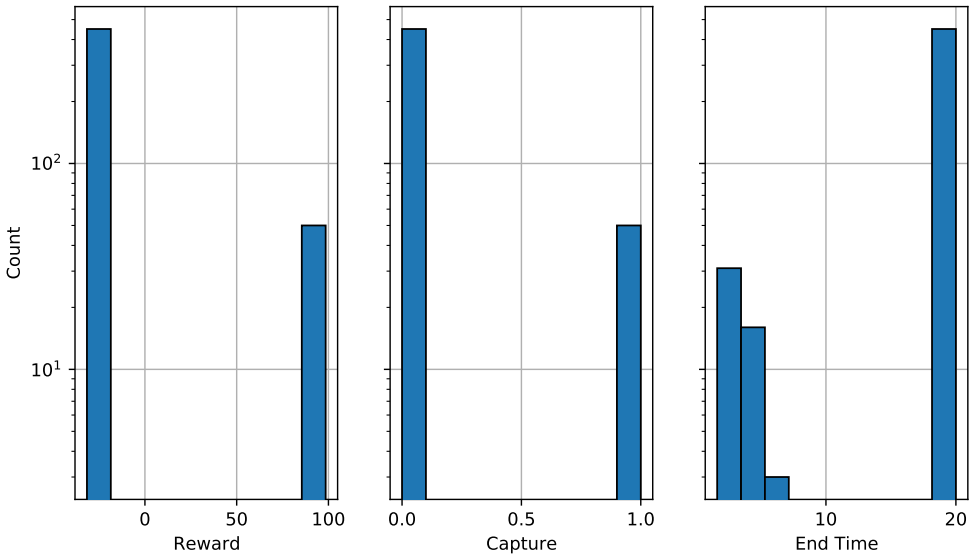


Figure 4.24: Histograms of best evader metrics against pursuer using PP

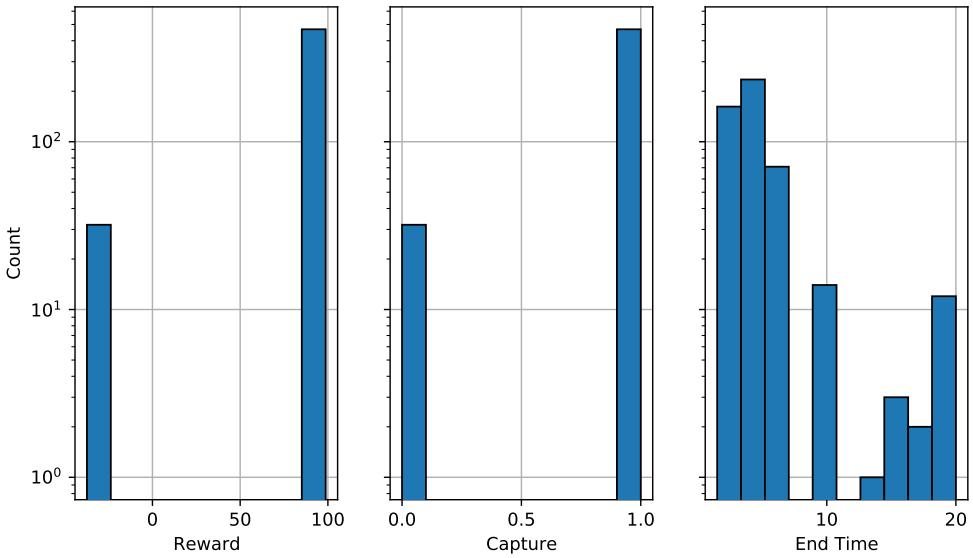


Figure 4.25: Histograms of best evader metrics against pursuer using PN

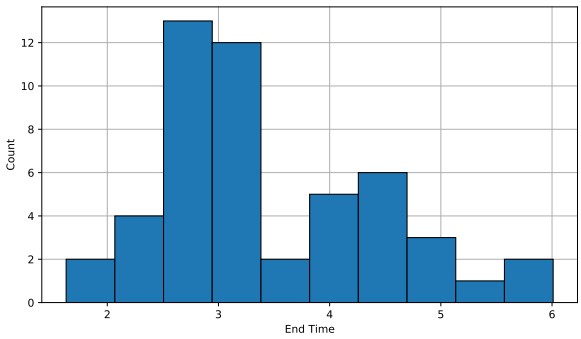


Figure 4.26: Distributions of end time when captured by pursuer using PP

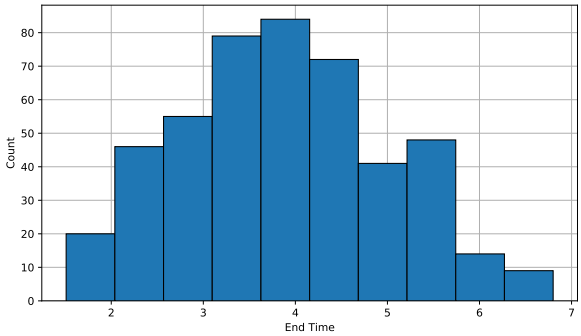


Figure 4.27: Distributions of end time when captured by pursuer using PN

Table 4.13: Results from applying TOPSIS to evaders using multiple sets of criteria

Criteria	**Rank 1**		**Rank 2**		**Rank 3**	
	Index	s_b	Index	s_b	Index	s_b
Capture	7	0.0844	20	0.0868	19	0.0885
End Time	22	0.3914	5	0.4033	11	0.4113
Reward	7	0.1192	20	0.1224	19	0.1237
Final	*7*	*0.1634*	*20*	*0.1650*	*19*	*0.1651*

with corresponding histograms shown in Figures 4.28 and 4.29. The tests showed the capture rate of the ANN-controlled evader was less than that for an evader using PE by very significant margins, independent of pursuer guidance. The ANN guidance was better than BE against pursuers using PN, but not so against pursuers using PE.

Table 4.14: Results of capture rate testing for ANN-controlled evaders

Baseline	**Pursuer**	$\bar{x}_0$	s_0	$\bar{x}_1$	s_1	t	ν
PE	PP	0.912	0.0284	0.099	0.0302	-1385	9959
BE	PP	0.080	0.0272	0.099	0.0302	33.44	9887
PE	PP	0.970	0.0173	0.937	0.0247	-77.67	8939
BE	PN	0.960	0.0196	0.937	0.0247	-52.79	9499

The pairwise capture test for each geometry was performed next. Counts for the four outcomes are reported in Table 4.15. Outcomes 3 and 4 were of particular interest since these were non-neutral results. The baseline guidance algorithms were able to avoid capture in 32 cases where the ANN was not. However, the ANN was able to avoid capture in 457 cases where a baseline algorithm could not. The latter was heavily skewed by the poor performance of PE versus PP. Compared to BE only, Outcome 3 was realized 13 times while Outcome 4 was realized 15 times. This indicated neither held a distinct advantage over the other in terms of effectiveness.

The data on Evader 7 was classified according to the types of outcomes identified previ-

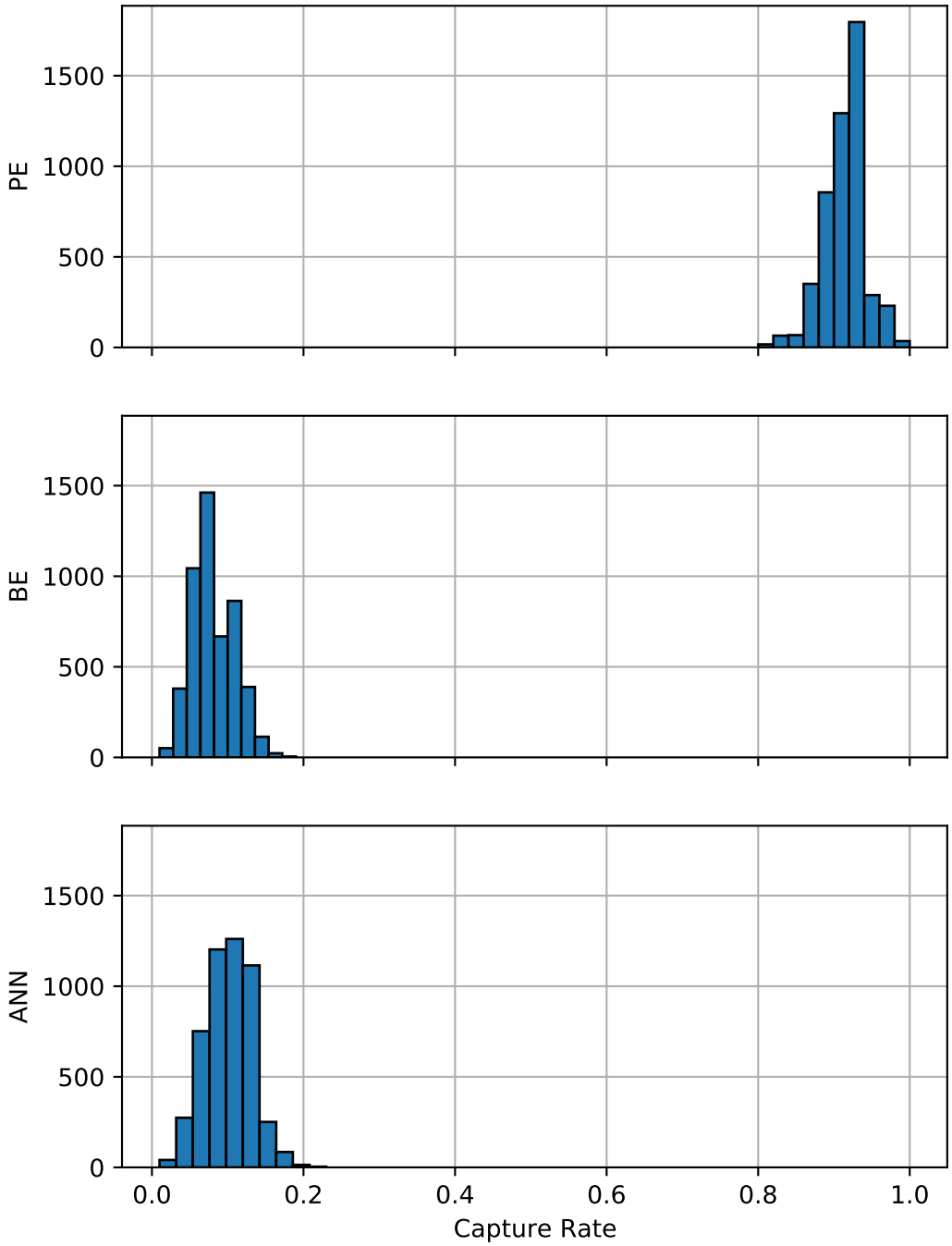


Figure 4.28: Distribution of estimated capture rates against pursuers using PP

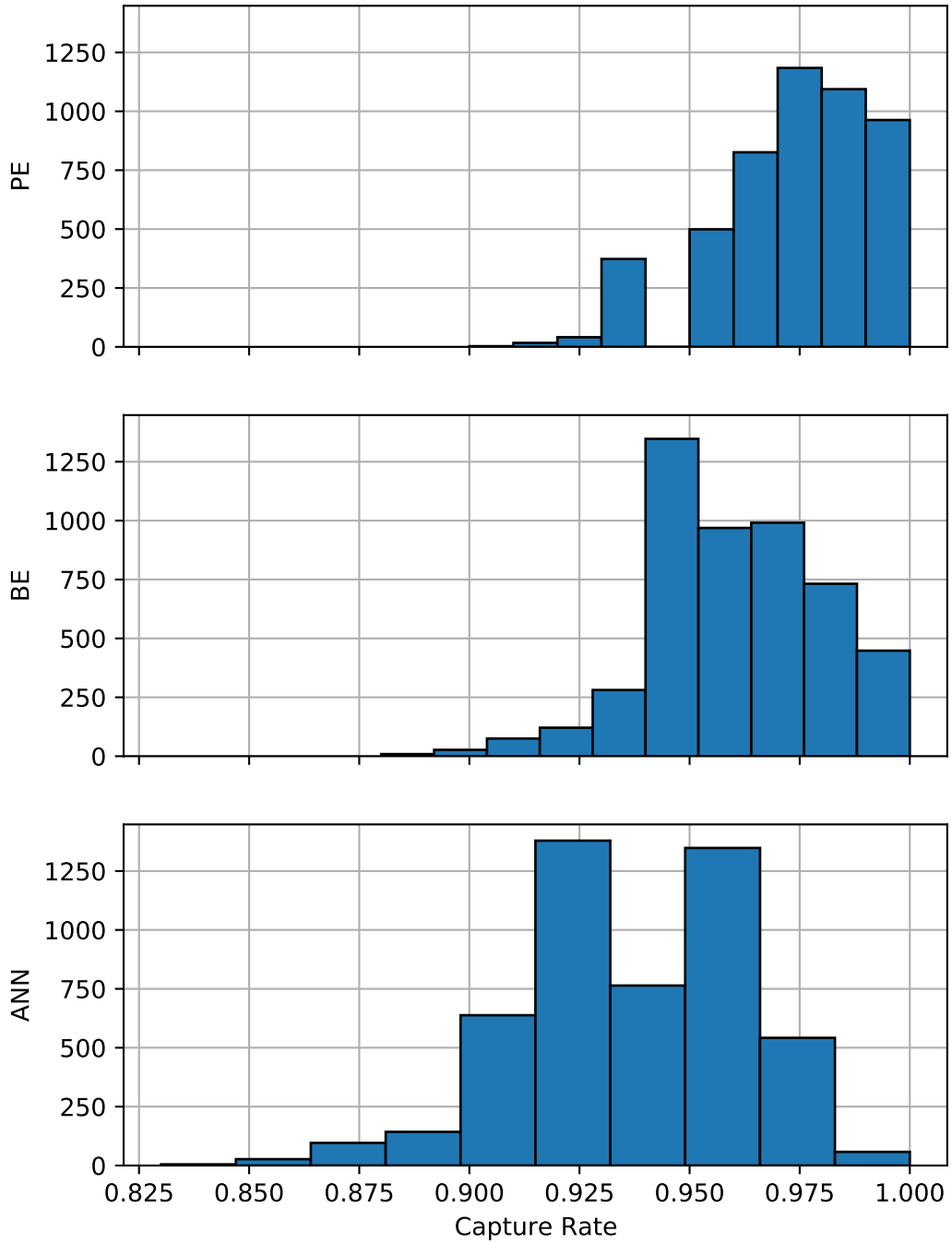


Figure 4.29: Distribution of estimated capture rates against pursuers using PN

Table 4.15: Results of pairwise capture test between evaders

Baseline	Pursuer	Outcome 1	Outcome 2	Outcome 3	Outcome 4
PE	PP	35	29	15	421
PE	PN	464	11	4	21
BE	PP	37	447	13	3
BE	PN	468	20	0	12
Total		*1003*	*507*	*32*	*457*

ously, the results of which are given in Table 4.16. The data show the evader selected using TOPSIS performed very well against pursuers using either baseline guidance algorithm. It avoided capture by a pursuer using PP in 90% of test cases, and achieved an average reward on par with evaders using BE. It also fared very well against a pursuer using PN.

Table 4.16: Performance statistics for the best evader on 500 test geometries grouped by capture and end time thresholds

Case	Outcome	Count	Reward		End Time	
			$\bar{x}$	s	$\bar{x}$	s
vs PP	1	50	96.1241	1.5822	3.4602	1.0121
vs PP	2	0	–	–	–	–
vs PP	3	0	–	–	–	–
vs PP	4	0	–	–	–	–
vs PP	5	450	-30.4129	0.5521	20.0000	0.0000
vs PN	1	468	94.9655	2.0400	3.8860	1.1436
vs PN	2	0	–	–	–	–
vs PN	3	6	-25.4957	0.2307	9.9017	0.0538
vs PN	4	14	-29.9347	4.9160	12.5764	2.9036
vs PN	5	12	-31.5629	0.7569	20.0000	0.0000

Two-sample t tests were conducted on the data grouped by outcome, the results of which are shown in Table 4.17. Several of the tests had high significance, as indicated by the last column. However, the results themselves were more nuanced. The ANN per-

formed worse than all of the baselines when captured in under 10 seconds, as indicated by the positive t values for reward and negative t values for end time. The results were more significant when comparing to PE than when comparing to BE. However, the ANN performed significantly better than all baselines when avoiding capture for 20 seconds.

Table 4.17: Results of statistical testing between evader groups

Baseline	**Pursuer**	**Outcome**	**Metric**	t	ν	p
PE	PP	1	Reward	6.4945	1.45e+06	<0.001
PE	PP	1	End Time	-7.2605	1.05e+06	<0.001
PE	PP	5	Reward	-15.7027	1.27e+05	<0.001
PE	PN	1	Reward	6.0874	2.11e+08	<0.001
PE	PN	1	End Time	-10.1984	2.08e+08	<0.001
PE	PN	3	Reward	-1.8758	123	0.063
PE	PN	3	End Time	1.1177	72	0.267
PE	PN	4	Reward	-1.0355	296	0.301
PE	PN	4	End Time	0.1963	313	0.845
PE	PN	5	Reward	-3.1942	1447	0.001
BE	PP	1	Reward	0.1216	1.59e+05	0.903
BE	PP	1	End Time	-0.7170	1.64e+05	0.473
BE	PP	5	Reward	-58.7826	1.87e+08	<0.001
BE	PN	1	Reward	0.9771	2.13e+08	0.328
BE	PN	1	End Time	-1.9951	2.12e+08	0.046
BE	PN	4	Reward	-1.6984	3244	0.089
BE	PN	4	End Time	0.5898	3845	0.555
BE	PN	5	Reward	-2.4113	1161	0.016

Summary

The results of statistical tests performed on the data generated by the training process for evaders indicated the models were able to learn effective behaviors for evading capture by a pursuer using either PP or PN. The best models performed at least as well as the two baselines, and the majority of these findings were statistically significant.

As with the pursuers, a final application of TOPSIS was used to rank all the 26 models against one another. The top-ranked evader ANN maintained its position as the most desirable alternative with a similarity of 0.1376; BE ranked 12$^{\text{th}}$ with a similarity of 0.1448; and PE ranked last with a similarity of 0.7742.

4.6.5 Comparing Trained Models

A final set of tests was performed by simulating each trained pursuer model against each trained evader model. The results of these simulations would help in determining how well the models were able to generalize their behaviors, since the agent models were not exposed to one another during training, as well as to establish a baseline for later experiments.

There were $24 \times 24 = 576$ combinations of pursuer and evader models. Each combination was simulated on the 500 test geometries and the metrics were recorded. TOPSIS was applied again, using the capture and end time metrics against each opponent over each geometry as criteria. The resulting rankings and similarity metrics are reported in Table 4.18. The last two columns report the rank and similarity of the model used in the respective analyses earlier in this section.

Pursuer 21, which was in the top three against PE and BE, was the top-ranked pursuer against all evaders, while the previously top-ranked Pursuer 0 was relegated to rank 17. Evader 20, which was ranked second against PP and PN overall, was the top-ranked evader against all pursuers. Evader 7 was relegated to 9$^{\text{th}}$ place with a similarity of 0.4488.

Table 4.18: Results from applying TOPSIS to competing ANN data

Agent	**Rank 1**		**Rank 2**		**Rank 3**		*Previous Best*	
	Index	s_b	Index	s_b	Index	s_b	Rank	s_b
Pursuer	21	0.1256	4	0.1294	6	0.1399	*17*	*0.2935*
Evader	8	0.4451	0	0.4466	20	0.4472	*8*	*0.4488*

Specific results for each pair of pursuer and evader models were collected and analyzed. Data were first grouped by outcome and the results are given in Table 4.19. It was seen that

Outcomes 3 and 4, which correspond to range escape, never occurred. Pursuer 21 appeared to perform significantly better than Pursuer 0 at capturing the evaders. However, neither achieved a level of effectiveness comparable to PN, which had 468 captures against the best evader. These results suggested the trained models could be effective against each other. However, they also exposed potential risks in not allowing the agents to learn simultaneously: Models which had previously performed very well in testing were shown to have flaws.

Table 4.19: Performance statistics for the best models on 500 test geometries grouped by capture and end time thresholds

Case	**Outcome**	**Count**	**Reward**		**End Time**	
			$\bar{x}$	s	$\bar{x}$	s
P21 vs E8	1	420	94.9168	2.1495	4.0008	1.2266
P21 vs E8	2	54	71.5471	1.5842	18.1026	0.8658
P21 vs E8	3	0	–	–	–	–
P21 vs E8	4	0	–	–	–	–
P21 vs E8	5	26	-30.3540	0.7230	20.0000	0.0000
P21 vs E7	1	421	94.9169	2.1469	4.0000	1.2255
P21 vs E7	2	54	71.6337	1.6174	18.0665	0.8769
P21 vs E7	3	0	–	–	–	–
P21 vs E7	4	0	–	–	–	–
P21 vs E7	5	25	-30.3466	0.7506	20.0000	0.0000
P0 vs E8	1	338	94.8732	2.2486	4.0612	1.2846
P0 vs E8	2	24	70.1516	1.5143	18.3717	0.7365
P0 vs E8	3	0	–	–	–	–
P0 vs E8	4	0	–	–	–	–
P0 vs E8	5	138	-31.0591	0.7036	20.0000	0.0000
P0 vs E7	1	344	94.8604	2.2515	4.0678	1.2860
P0 vs E7	2	25	70.2823	1.5354	18.3228	0.7799
P0 vs E7	3	0	–	–	–	–
P0 vs E7	4	0	–	–	–	–
P0 vs E7	5	131	-31.1941	0.6612	20.0000	0.0000

4.6.6 Conclusions

The results of this experiment showed that an application of reinforcement learning can produce behavior models which perform at least as well as off-the-shelf baselines for scenarios with multiple agents in competition with one another. **The data produced by this experiment supported Hypotheses 1 and 2** because the models were able to achieve high levels of performance and effectiveness, even in the face of delayed rewards and uncertain responses by their opponent. Further, few assumptions had to be made about to the form of the state-action mapping and behaviors similar to the mathematically intensive PN guidance algorithm, which involves several vector cross products, were still achieved.

Evaluation of the trained ANNs against one another and subsequent analyses indicated the need for a more robust training procedure in competitive scenarios where agent interactions can drive outcomes. This helped to substantiate the gaps identified earlier and the need for further experimentation.

4.7 Experiment 2: Multi-Agent Reinforcement Learning

A key challenge in multi-agent scenarios is the massive space of possible paths to be explored. The average duration of the test simulations run in the previous experiment where the evader was captured was approximately 4 seconds. The two-player pursuit-evasion scenario where each player can take one of three actions 20 times per second and running for 4 seconds yields a maximum $9^{20\times4} \approx 10^{76}$ possible decision paths. However, only one agent was actively experimenting with alternative decision-making processes to explore that space – the other was using a known, deterministic algorithm. However, known models of behavior may not always be available and, in such cases, it would be necessary to employ a methodology with a trusted ability to explore the space of possible behaviors.

The second experiment was designed to test Hypothesis 3, regarding the use of MARL in the proposed methodology. It was hypothesized that creating multiple ANNs per agent

and then randomly grouping those models at the start of each iteration of simulation and training would allow for effective explorations of the behavior space. This experiment was designed to test this hypothesis by implementing both the identified training processes and comparing them in several ways to determine how well the models would be able to develop robust, general behaviors. The results of this experiment would help to substantiate Hypothesis 3 if the models produced by application of MARL using multiple models per agent performed at least as well as those produced using only a single model per agent against the baselines, as well as against each other.

4.7.1 Training Procedure

The overall training procedure for this experiment was very similar to that for the first experiment. Several hyperparameters were modified based on the results of the first experiment. Most notably, the number of episodes was reduced from 50,000 to 20,000.

Two distinct training cases were implemented. The first trained pairs of models – one for the purser and one for the evader – against each other for the entirety of the training process, and this was repeated 24 times. This was intended as an analog for the approach of decentralized execution with centralized training. However, the decentralized execution was largely omitted. This was deemed acceptable because the team hide-and-seek environment used by Baker et al. was much more complex than the pursuit-evasion scenario. This case will henceforth be referred to as the **Individual** process.

The second case used a completely decentralized approach to training the models. Twenty-four models were initialized for each agent and were randomly paired at the start of each episode for simulation and training. This case will henceforth be referred to as the **Population** process. The architectural difference between the training schemes used in these two cases is shown in Figure 4.30.

In both cases, the models were only trained against other ANN-controlled agents which were also learning. At no point during the training were the models exposed to the baseline

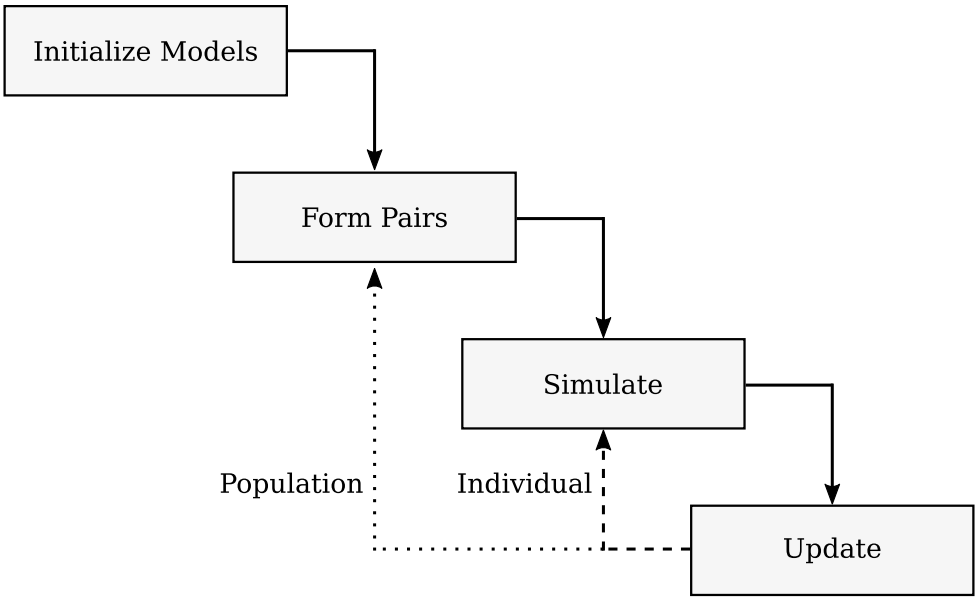


Figure 4.30: Comparison of test cases for multi-agent reinforcement learning

guidance algorithms for training purposes. This was done to test the capacity for models to learn *tabula rasa*, as might be the case in situations where models of behavior are not readily available.

4.7.2 Testing the Processes

Testing the processes was done in the same manner as was used for the first experiment. Models were tested at regular intervals throughout the training process. However, the models were only tested against the baseline guidance algorithms and not against each other. This was done because the model-versus-model testing would have been more difficult to interpret, since low pursuer rewards could result from a poor pursuer model or a good evader model. Using the baselines removed this uncertainty and allowed for better comparisons of performance. It also provided data to determine how well the models had generalized their respective policies since they were not trained against the baselines.

All trained models were simulated against the baseline opponent guidance algorithms on the 500 test geometries after training had completed in order to compare performance on a statistical basis. Testing against the baseline guidance algorithms allowed for fair com-

parisons between the training processes. Akin to the zeroth law of thermodynamics, if one training process produced models which were statistically better than the other against an adversary behaving the same way then that process could be considered superior. Controlling for the adversary behaviors is important in reducing confounding factors when making such comparisons.

A model was to be selected for each agent from each training process for testing against the selected model for the opposing agent from the other process. That is, the best Individual-trained pursuer would be tested against the best Population-trained pursuer, where "best" is determined by applying TOPSIS to intra-process competitive performance evaluations. The testing procedure is depicted in Figure 4.31. The purpose of this test was to allow the processes to be compared against one another head-to-head. The selection of a best model is needed to support the overarching methodology; this test would aid in determining which training process would be more likely to produce a more effective model of behavior and, by extension, provide better support for the overarching methodology.

4.7.3 Training Results

Individual-Trained Models Versus Baselines

Figure 4.32 shows the trends in average capture over the five test geometries versus training episode for the pursuer and evader models, respectively. The trends for the pursuer were similar to those from the first experiment, with each the average model improving as training progressed. At least one model was able to capture evaders using either guidance algorithm in each of the five test geometries starting around episode 7,000, suggesting the models had learned effective pursuit policies.

The evader models trained using this process appeared to have a wide variation in performance against pursuers using the baseline guidance algorithms. At least one evader was able to do well against pure pursuit, but the average evader was unable to avoid capture in more than two of the five test geometries. The model performance against proportional

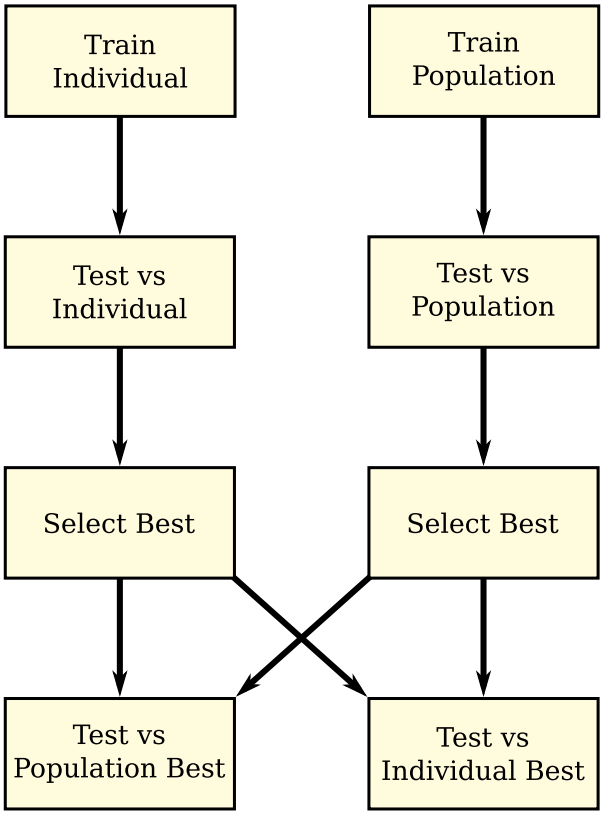


Figure 4.31: Inter-process testing procedure

navigation followed a similar trend as was seen in the first experiment.

Population Training Versus Baselines

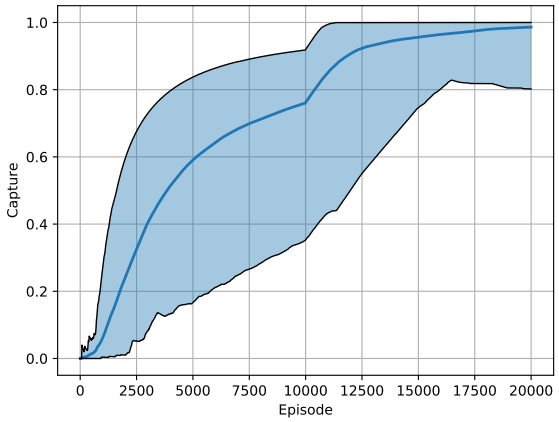
Trends in performance against the baseline guidance algorithms for the pursuer and evader models trained using the population training process are shown in Figure 4.33. Trends in pursuer performance were similar to those from the individual training process. However, at least one model appeared to perform poorly against beam evasion. The upward trend in the lower bound towards suggests additional training might have allowed the model to perform better.

The trends in evader performance indicated better performance than the individual process was able to achieve. The average capture over the five geometries for both pursuer guidance algorithms was lower for the population-trained models than for the individual-trained models. At least one of the evaders saw significant degradation in performance against proportional navigation starting around episode 5,000. However, this had little effect on the average performance over all 24 models, suggesting the other 23 were much closer to the lower bound shown.

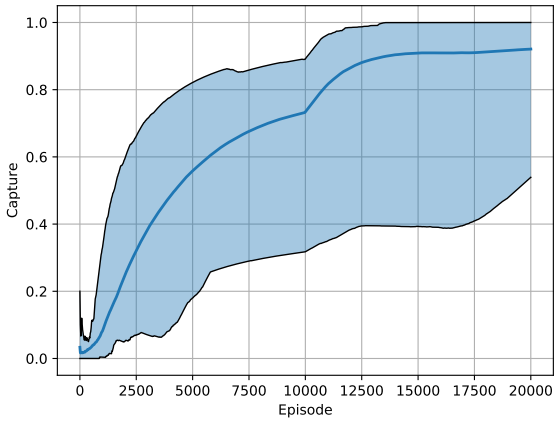
4.7.4 Statistical Testing Versus Baselines

All of the trained models were tested against the baseline guidance algorithms for their opponent in order to make fair comparisons between the training processes. This allowed the two processes to be evaluated against known behaviors and establish a baseline for further analysis. Distributions of the performance metrics for each of the eight cases are given in Appendix B.

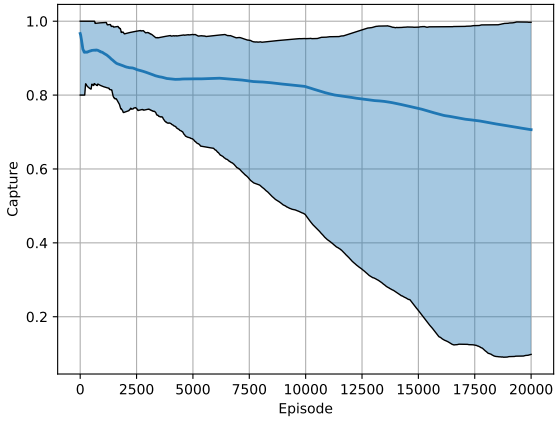
The models were simulated against each of the two baseline opponent guidance algorithms on the 500 test geometries and their metrics were recorded. Statistical data on the performance of the fully-trained pursuer models against evaders using baseline guidance algorithms are given in Table 4.20. The data reported are the average and standard de-



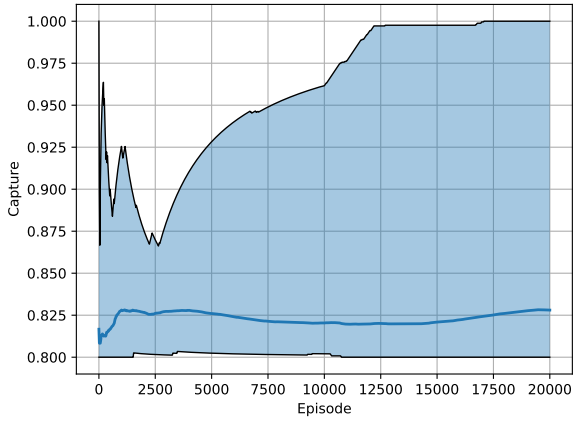
(a) Pursuers vs PE



(b) Pursuers vs BE

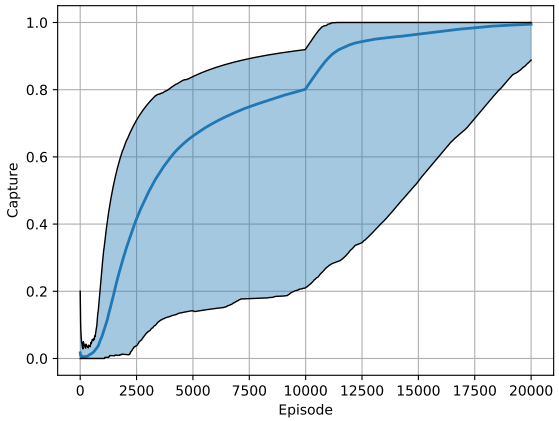


(c) Evaders vs PP

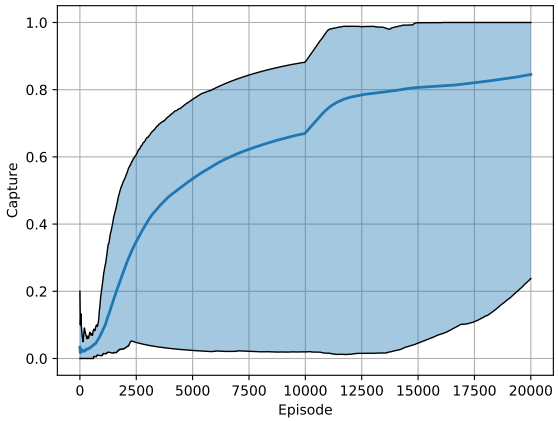


(d) Evaders vs PN

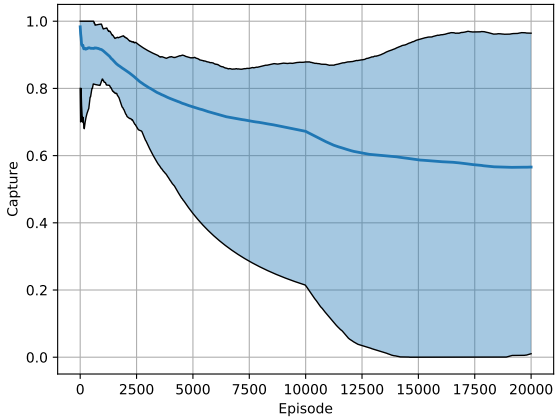
Figure 4.32: Trends in average capture versus episode for individual-trained models against baseline opponents



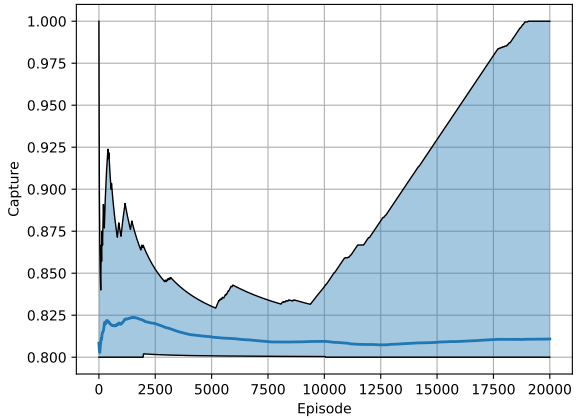
(a) Pursuers vs PE



(b) Pursuers vs BE



(c) Evaders vs PP



(d) Evaders vs PN

Figure 4.33: Trends in average capture versus episode for population-trained models against baseline opponents

viation of each metric over all 24 models trained using the two processes. Overall, both training processes could be expected to produce effective models. The Population pursuers did better against evaders using PE while the Individual pursuers did better against evaders using BE. However, the Population pursuers had high variances against BE, suggesting some models may have been able to perform well.

Table 4.20: Performance statistics for fully-trained pursuers vs baseline

		Individual		Population	
	Evader	$\bar{x}$	s	$\bar{x}$	s
Pursuer Reward	PE	88.9157	23.8386	90.6047	19.7724
	BE	89.0286	23.5906	83.1723	35.4480
Capture	PE	0.9633	0.1879	0.9748	0.1569
	BE	0.9642	0.1859	0.9089	0.2877
End Time	PE	5.5729	3.9569	5.3352	3.0726
	BE	5.5537	3.9271	5.8056	4.9701

Statistical data on the performance of the fully-trained evader models against pursuers using baseline guidance algorithms are given in Table 4.21. Similar to the pursuers, the data on the evaders shows both processes produced effective models. However, the Population process appeared to have produced more effective models than the Individual one based on these statistics.

Table 4.21: Performance statistics for fully-trained evaders vs baselines

		Individual		Population	
	Pursuer	$\bar{x}$	s	$\bar{x}$	s
Pursuer Reward	PP	35.7965	61.7849	25.5266	61.6803
	PN	90.2138	23.0993	89.3650	25.1732
Outcome	PP	0.5338	0.4989	0.4500	0.4975
	PN	0.9642	0.1859	0.9568	0.2032
End Time [s]	PP	12.0648	7.6162	13.2515	7.6094
	PN	4.5148	2.1770	4.5362	2.3268

Testing Training Process Effect

Tables 4.20 and 4.21 only allowed for high-level comparisons between the two training processes. More rigorous statistical analysis was warranted to determine if the choice of process significantly impacted expected model performance. Capture rate was selected as the dependent variable to facilitate testing; the other metrics had distributions which could make testing more difficult. The null hypothesis was $H_0 : p_0 = p_1$ and the alternative $H_1 : p_0 \neq p_1$, where p_0 corresponds to the capture rate achieved by the Individual models and p_1 to that achieved by the Population models. The binomial test statistic (4.17) was calculated using the maximum likelihood estimate of the rate parameters for each case, where the number of samples $n_0 = n_1 = 24 \times 500 = 12,000$. The results are given in Table 4.22.

$$z = \frac{\hat{p}_0 - \hat{p}_1}{\sqrt{\hat{p}(1-\hat{p})\left(\frac{1}{n_0} + \frac{1}{n_1}\right)}}, \quad \hat{p} = \frac{n_0\hat{p}_0 + n_1\hat{p}_1}{n_0 + n_1} \tag{4.17}$$

Table 4.22: Binomial test results for MARL training processes versus baselines

Opponent Guidance	$\hat{p}$	z
PE	0.9690	-5.106
BE	0.9365	17.55
PP	0.4919	12.99
PN	0.9605	2.92

The results of the binomial test were all highly significant. The least significant result – evaders versus PN – had a p-value of 0.00175 on a one-tailed test. The null hypothesis could therefore be rejected in each case, and it could be concluded that the choice of training process does have a statistically significant impact on expected performance. However, the results confirmed the non-uniformity in performance against baselines observed previously: Population pursuers were less likely to capture an evader using PE. Note that, for the bottom

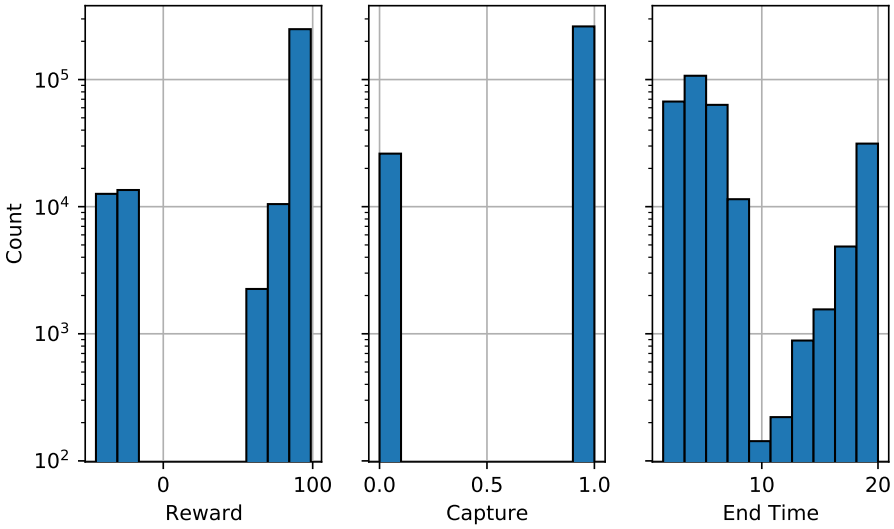
two rows corresponding to the evader models, a lower rate parameter was desirable. The data therefore indicate the Individual-trained evaders were captured *more often* than the Population-trained evaders, and therefore performed worse.

4.7.5 Statistical Testing Between Processes

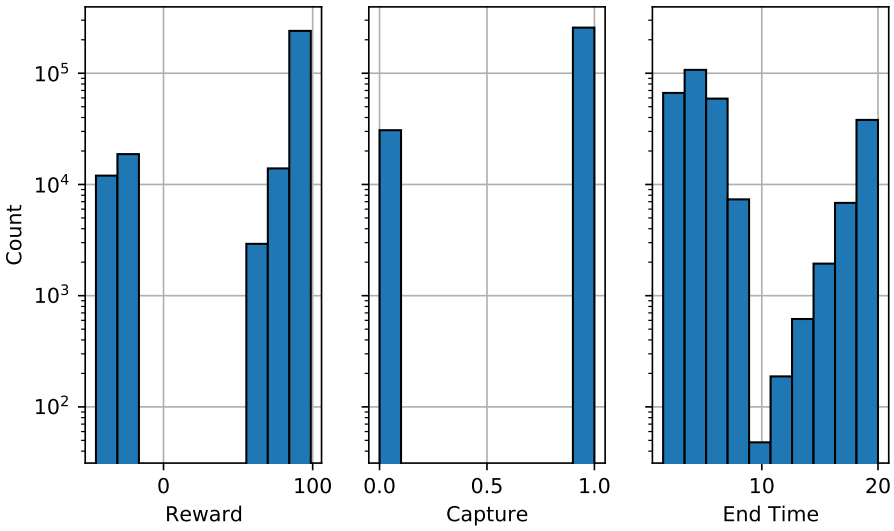
Performance statistics were collected for pursuer models were tested against evader models trained using the same process. Histograms of the reward, capture, and end time metrics are shown in Figure 4.34. The distributions for both processes appeared to show the pursuers were extremely effective, being able to capture the evaders far more often than not. The valley of end times around the 10 second mark indicates the same bimodality occurred as was seen in the first experiment, where the pursuer might miss an initial opportunity and then be forced to make a near-complete turn to re-engage.

Selection of Bests

TOPSIS was used to select a pursuer and evader model from each training process. As before, the capture and end time metrics for each of the 500 simulated geometries against each of the 24 intra-processes opponents were used as the criteria for TOPSIS. The results for both processes are given in Table 4.23. Notably, the top-ranked Individual models were not from the same pairings. That, while Pursuer 10 was the best among all pursuers in intra-process testing, Evader 10 ranked last with a similarity of 0.7435. On the other hand, whereas Evader 20 was the best among all evaders, Pursuer 20 ranked 13^{th} with a similarity of 0.2799. An exact cause of this phenomenon would be extremely difficult to identify because of how the training process is driven by interactions between the models, as well as random factors. However, these findings do suggest the Individual training process can fail to produce effective *sets* of models for multi-agent scenarios, even though single models from those sets may be very effective.



(a) Individual models



(b) Population models

Figure 4.34: Distributions of intra-process performance metrics

Table 4.23: TOPSIS results for intra-process metrics

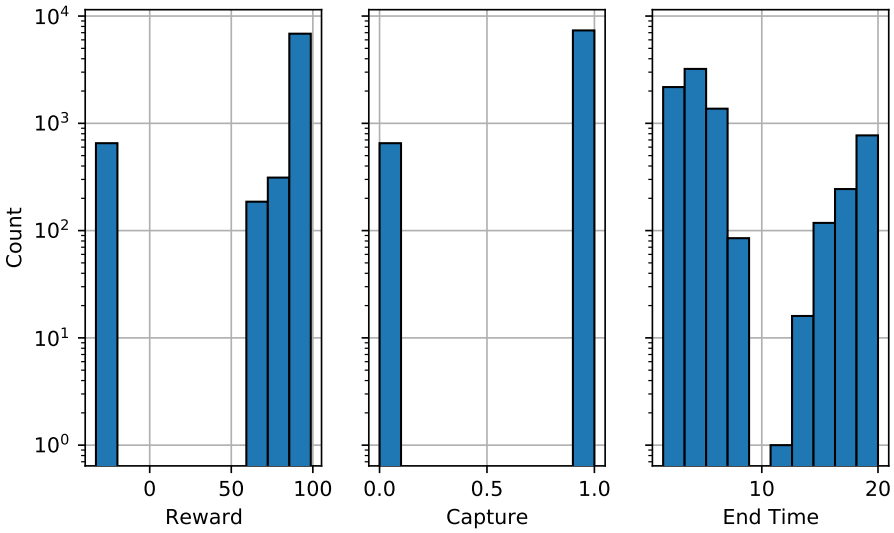
Process	Agent	Rank 1		Rank 2		Rank 3		Rank 4	
		Index	s_b	Index	s_b	Index	s_b	Index	s_b
Ind	Pursuer	10	0.1786	4	0.2078	12	0.2099	22	0.2279
	Evader	20	0.5529	13	0.5722	1	0.5781	8	0.5927
Pop	Pursuer	8	0.2216	13	0.2326	20	0.2357	6	0.2402
	Evader	8	0.4810	6	0.5088	23	0.5168	9	0.5243

Comparison of Bests

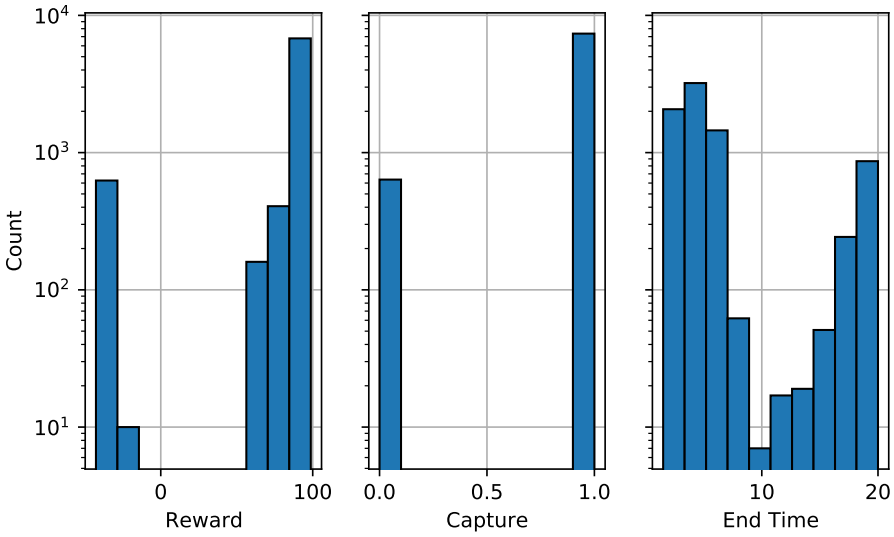
The top four models for each agent from each process were tested against an opponent model from the other process. Distributions of performance metrics for all combinations the top four pursuers from one process versus the top four evaders from the other are shown in Figure 4.35. These distributions indicated the Population pursuers were able to capture the Individual evaders in more cases than Individual pursuers could capture Population evaders. However, more detailed analysis was needed.

The data for each set inter-process tests was grouped by which of the five outcomes was realized for each simulation. These results, along with those for similar analyses conducted on the intra-process data, are given in Table 4.24, where the first two columns indicate the training process used for that agent: I for Individual or P for population. The grouped data did not identify one process as clearly superior to the other. Population pursuers captured evaders in fewer cases than Individual pursuers, regardless of which process the evader was from. Population pursuers took about as long to capture Individual evaders as Individual pursuers, but captured Population evaders slightly faster. Notably, range escape (Outcomes 3 and 4) occurred a total of 43 times for Population pursuers, suggesting the existence of some PN-like flaw in at least one of the pursuer models.

The data on Population pursuers was further analyzed in an attempt to identify the source of the Outcome 3 and Outcome 4 points. These analyses revealed all Outcome 3



(a) Individual pursuers versus Population evaders



(b) Population pursuers versus Individual evaders

Figure 4.35: Distributions of inter-process performance metrics

Table 4.24: Inter-process test data statistics grouped by outcome

Pursuer	Evader	Outcome	Count	Reward		End Time	
				$\bar{x}$	s	$\bar{x}$	s
I	I	1	7028	94.7320	2.2846	4.1361	1.3212
I	I	2	378	72.2437	2.5567	17.6764	1.4279
I	I	3	0	–	–	–	–
I	I	4	0	–	–	–	–
I	I	5	594	-30.4092	0.7858	20.0000	0.0000
I	P	1	7300	94.5919	2.3880	4.2601	1.4023
I	P	2	240	74.2129	2.6832	16.6897	1.6021
I	P	3	0	–	–	–	–
I	P	4	0	–	–	–	–
I	P	5	460	-30.6241	0.6915	20.0000	0.0000
P	I	1	6797	94.7474	2.2577	4.1398	1.3025
P	I	2	567	72.1551	2.8100	17.7417	1.5407
P	I	3	3	-25.1635	0.4408	9.4833	0.2558
P	I	4	21	-31.1412	4.7309	12.6662	3.2339
P	I	5	612	-30.6292	1.0243	20.0000	0.0000
P	P	1	6575	94.8044	2.2432	4.0832	1.2871
P	P	2	608	72.5246	2.4188	17.4970	1.3897
P	P	3	0	–	–	–	–
P	P	4	19	-34.4730	4.2882	15.2226	3.0051
P	P	5	798	-30.8377	0.8531	20.0000	0.0000

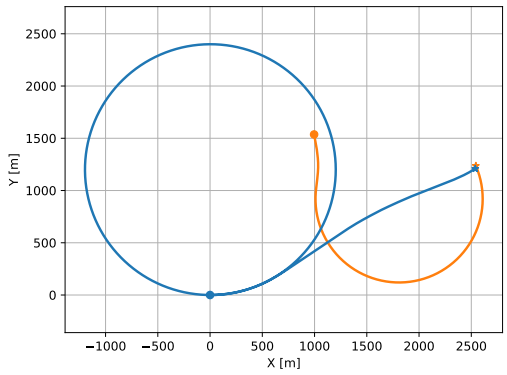
and 4 points were attributed to Pursuer 6, which was the fourth-ranked model. Pursuer 6 also accounted for 223 (36.4%) of the Outcome 5 data points against Individual evaders and 287 (35.9%) of those against Population evaders. In short, Pursuer 6 was making the Population process look far worse than it would have had only the top three models been considered. However, even with Pursuer 6 omitted, the Population pursuers performed worse than the Individual pursuers overall.

Top-Ranked Model Head-to-Head

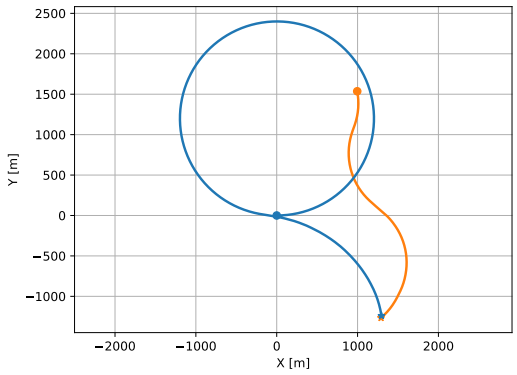
The top-ranked models for each agent from each process were tested against that for the opponent from the other process. Trajectories for each pair of models were generated and are shown in Figure 4.36. The maneuvers by the evaders are the distinguishing features. Both evaders force the pursuers to miss an early opportunity by maintaining a position inside the latter's turn radius. The Individual evader sustains a hard left turn after this initial miss, while the Population evader initiates a weaving maneuver. Neither is successful in avoiding capture a second time, but the Individual evader is able to delay capture by about 2 seconds because the sustained turn allows it to build some separation from the pursuer as the latter rounds the turn. The Population evader also breaks across the path of the pursuer in both cases, which maximize LOS rate but also increases closure rate.

The pursuer trajectories showed some slight differences between the two models. The Individual pursuer took a straighter line coming out of the turn against the Individual evader, as compared to that taken by the Population pursuer. The effect was small; the Individual pursuer achieved capture in 17.30 seconds and the Population pursuer in 17.41 seconds. Interestingly, the opposite occurred against the Population evader, where the Population pursuer broke off its turn earlier to close the distance with the evader faster. This had a greater impact on performance; the Population pursuer achieved capture in 15.34 seconds compared to the Individual pursuers time of 15.73 seconds.

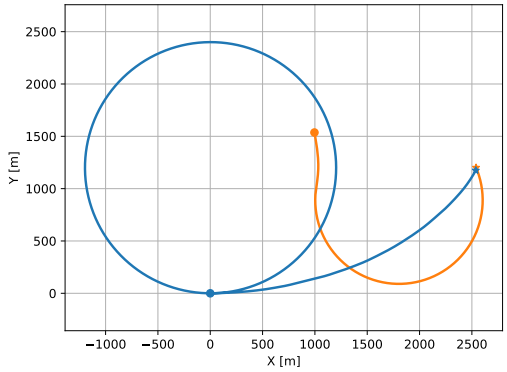
It should be noted that, in test geometry 2, it would be possible for the evader to avoid capture for the full duration of the simulation by maintaining its position inside the turn radius of the pursuer. This was demonstrated by Population pursuer 11 and Population evader 13, shown in Figure 4.37. However, this strategy runs somewhat counter to the objective function the evaders were attempting to maximize, since it requires maintaining a relatively low separation from the pursuer. It is not immediately clear if there exists a counter-strategy available to the pursuer which would mitigate this. If so, it would likely involve reversing the turn at some point, though it is possible there exists an evader strategy



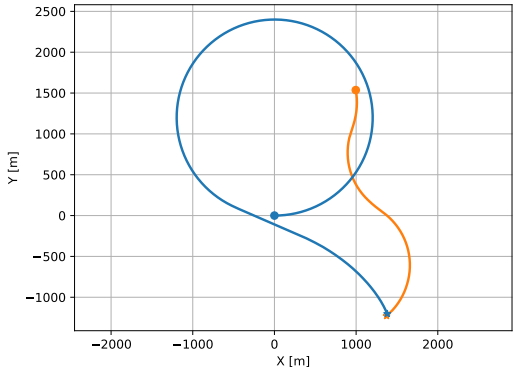
(a) Ind. Pursuer vs Ind. Evader



(b) Ind. Pursuer vs Pop. Evader



(c) Pop. Pursuer vs Ind. Evader



(d) Pop. Pursuer vs Pop. Evader

Figure 4.36: Trajectories of best pursuers vs best evaders on test geometry 2

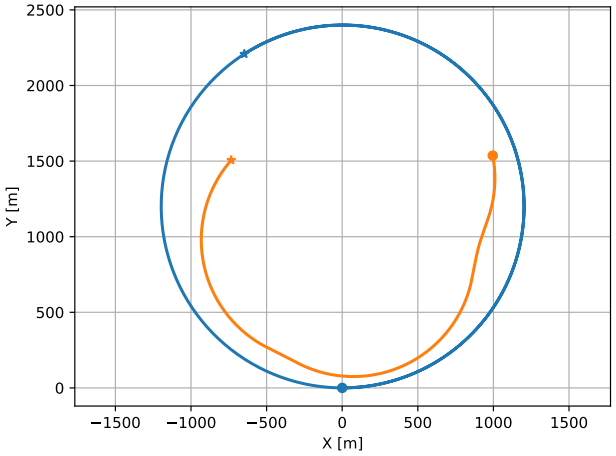


Figure 4.37: Possible winning strategy for evaders in test geometry 2

which could counter such a counter-strategy.

Top-Ranked Model Statistics

Data were collected for statistical analysis of inter-process performance for top-ranked models and are reported in Table 4.25, where p is the capture rate – i.e. the number of times the evader was captured as a fraction of the total number of simulations – and TTI is the average time to achieve capture. The calculations of TTI only included cases where the evader was captured.

Table 4.25: Inter-process performance metrics for best models

P \ E	**Ind**	**Pop**
Ind	$p = 0.934$ $TTI = 4.689$	$p = 0.922$ $TTI = 4.518$
Pop	$p = 0.944$ $TTI = 5.483$	$p = 0.878$ $TTI = 4.566$

The performance statistics on the best-performing models are mixed. In terms of capture rate p, Population models performed better in the inter-process test: The Population pursuer captured the Individual evader more often than the Individual pursuer, and the Population evader avoided capture by the Individual pursuer in more cases than the Individual evader. However, the Population pursuer took longer to capture the Individual evader and the Population evader was captured faster, on average.

Splitting the data into captures in under 10 seconds and those in over 10 seconds was insightful, although not altogether helpful. When capturing the Individual evader in under 10 seconds, the Individual pursuers took an average of 4.155 seconds over 448 cases, while the Population pursuer was slightly faster at 4.069 seconds over 423 cases. However, when capturing said evader in more than 10 seconds, the Individual pursuer took an average of 17.268 seconds over 19 cases and the Population pursuer took 17.694 seconds over

49 cases. Overall, the Population pursuer had more captures but took longer; if effectiveness were more important than performance than the Population approach would likely be preferred for pursuers.

The evader results were similarly mixed: The Population evader had lower capture rates against, but was also captured faster by both pursuers. Metrics were similar between both pursuers against the Population evader: Against the Individual pursuer, 445 captures with an average TTI of 4.082 seconds and 16 with an average of 16.649 seconds; against the Population pursuer, 420 captures with an average time of 4.012 seconds and 19 captures at an average of 16.809 seconds. The main difference was the lower overall capture rate against the Population pursuer.

4.7.6 Hypothesis Testing

Hypothesis 3 was tested by collecting all the data on the top-performing models for an application of TOPSIS. Capture and end time metrics from the 500 test simulations were used as criteria, and the baseline guidance algorithms were included for completeness. However, this meant the pursuers and evaders had to be evaluated separately, since the ANN-controlled pursuers versus baseline evaders would be incomparable to the ANN-controlled evaders versus baseline pursuers. The resulting rankings and similarities are given in Table 4.26.

Table 4.26: Overall TOPSIS results

Agent	**Rank**	**1**	**2**	**3**	**4**
Pursuer	*Model*	PN	Ind	Pop	PP
	Similarity	0.1266	0.1927	0.2361	0.8639
Evader	*Model*	Pop	Ind	PE	BE
	Similarity	0.3338	0.4325	0.6848	0.7186

Proportional navigation was the best option for the pursuer, and pure pursuit was the worst by a wide margin. The two training processes for MARL were relatively close to-

gether by comparison, with the Individual process holding an advantage. However, the Population process was the best option for the evader while the Individual process was a distant second. The baseline evasion algorithms were decidedly poor alternatives compared to the trained ANNs. The results did not strongly support Hypothesis 3, but neither did they refute it. Both processes for MARL were effective for exploring behavior spaces and identifying high-performing policies. This finding agreed with the largely unsubstantiated claim by Baker et al. that both processes could produce similar results to one another, but more rigorous analysis and evidence was presented here. Ultimately, **the data indicated the use of multiple models per agent to be a feasible approach to training artificial neural networks as behavior models in multi-agent scenarios where existing models of behavior are not available.**

Closer examination of the training partners for the top-ranked Individual-trained models helped to substantiate the hypothesis. Evader 10, whose partner was the top-ranked Individual pursuer, had an average capture rate of 0.99 against the top-ranked Population pursuer with an average time of 5.19 seconds. Pursuer 20, whose partner was the top-ranked Individual evader, had an average capture rate of just 0.782 against the top-ranked Population evader, with an average time of 5.57 seconds. These results indicated the strength of the Individual training process could only be realized by running it several times, ideally in parallel, as any single run would be unlikely to produce both an effective pursuer *and* an effective evader. When collected into tabular formats similar to Table 4.25 where the Individual cells are populated by training pairs, the Population process strongly dominated the Individual process. Taken together, **these results suggested the Population process could be superior to the Individual process when computational resources are available.**

4.8 Experiment 3: State Space Augmentation

The third experiment was designed to test Hypothesis 4 and attempt to address Gap 1 regarding the incorporation of design attributes into the exploration of employment concepts

using MARL. It was hypothesized that design variable settings could be included into the state vector, and that the behavior models would be able to leverage this information in their decision-making processes.

The hypothesis was to be tested by constructing and training two sets of models, one with design variables included in the state space and the other without. Training for each would proceed identically, and the performance of the two sets would be compared at the end to determine whether or not the inclusion of design variables in the state space improved performance. The two sets of models were constructed for both pursuers and evaders but, as in Experiment 1, they were only trained against the baseline guidance algorithms. MARL was not used in this experiment in order to improve control over the process and isolate the effect of state space augmentation on performance.

4.8.1 Implementation

There were three primary implementation questions which had to be addressed in the conduct of this experiment. The first was: How should design variables be sampled for simulation and data generation? Two alternatives were identified: A space-filling DOE or random sampling. A space-filling DOE would ensure the models were able to experience design variable settings which covered the design space, possibly allowing them to learn more effective behaviors. However, this could also significantly increase the computational cost to perform the analyses if there were a large number of points sampled with the DOE.

Random sampling does not guarantee coverage of the design space but would likely be less of a burden on computational resources. The same training procedure as was used in previous experiments could be employed here, where the agents were simulated until they had generated a threshold number of samples. Each simulation in that process would have a different, randomly sampled set of design attributes associated with it. Coverage of the design space over the course of the entire training process would be almost certain by virtue of the large number of episodes. For these reasons, random sampling was the

approach chosen for this experiment.

The second implementation question was: How should the design variables be represented in the state space? They could be passed to the network without any transformation, which would be easiest, or they could be linearly transformed in some manner. In theory, there would be no difference because the operations inside the ANN contain linear transformations; multiplying the input by a constant and dividing the weight by that same constant results in no net difference. However, in practice, it may be difficult for the network to learn the different influences of design variables if they are not standardized in some way. In the design problem here, the two design attributes under consideration differ by three orders of magnitude, and that may hinder the training process. Furthermore, the exact value of the state variable should make no difference to the network; what matters is *where* in the design space it is. By this reasoning, the linear transformations (4.18) were used.

$$\bar{u}_e = -1 + 2\frac{u_e - u_{min}}{u_{max} - u_{min}}, \quad \bar{\omega}_e = -1 + 2\frac{\omega_e - \omega_{min}}{\omega_{max} - \omega_{min}} \tag{4.18}$$

The last implementation question pertained to how the ANNs would be modified to accommodate the expanded state space. There were many possible approaches to this. The first considered was to simply expand the state space, treating the design variable settings as observable states, and leave the rest of the network as-is. However, preliminary tests showed this to be an unreliable method for augmenting the state space. It was hypothesized that the network was unable to adequately "understand" the influence of the design variables on the evolution of the environment using this simplistic approach.

It was hypothesized, based on the preliminary results, that separating the states observed from the environment from the design variables would yield better results. However, there were still many ways to do this, each a variation of how the variations elements of the network were connected. Some examples are shown in Figure 4.38.

The chosen ANN architecture was only a slight modification from the basic one. The state space vector was split into two channels for the first hidden layer. One channel handled

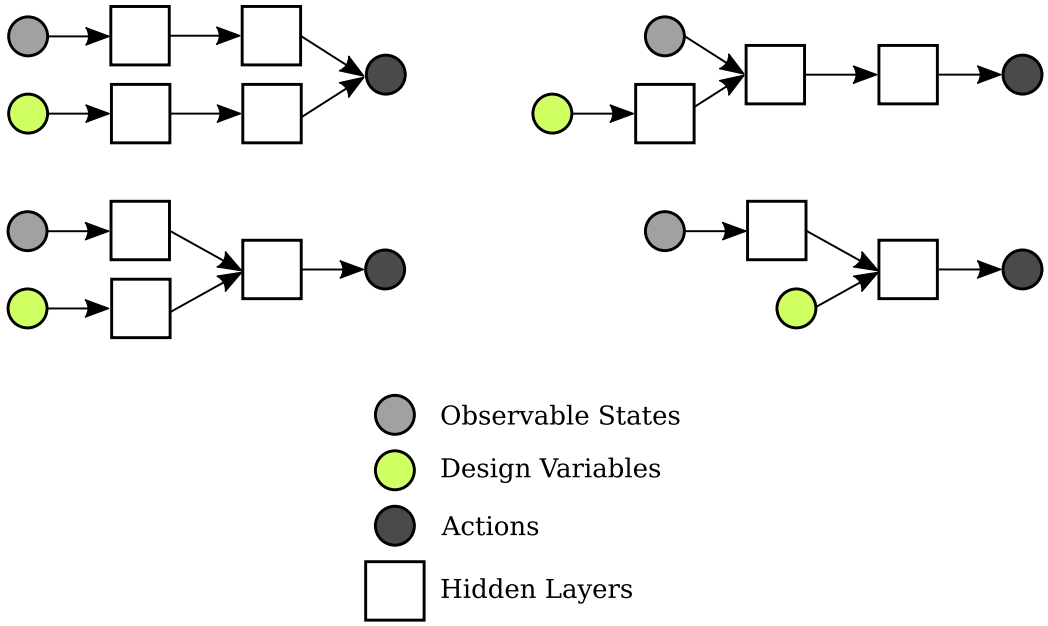


Figure 4.38: Possible network architectures for augmented state spaces

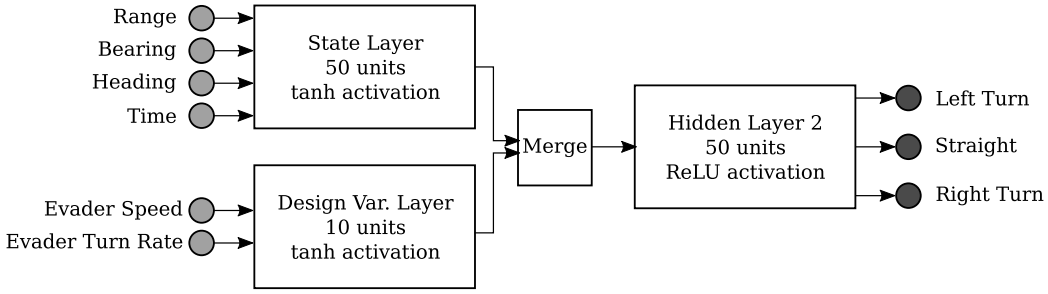


Figure 4.39: Modified neural network architecture

the original state space and the other handled the design variable settings. The former had 50 nodes, while the latter had 10. Both channels used the hyperbolic tangent activation function. The output from both channels was concatenated before being passed through the remainder of the model, the details of which were not modified from previous experiments. The architecture is shown in Figure 4.39.

4.8.2 Testing the Models

The hypothesis that augmenting the state space with design variable settings would result in models which were better able to leverage or mitigate the effects of those variables on the scenario would be tested by comparing the performance between the two types of models.

A standard testing procedure was established by generating a 1,000-point LHS over the design space and 50 test geometries. Each design point would be used to simulate models on the 50 geometries, allowing for control over the influence of both factors simultaneously. The two sets of models could then be compared by their performance on each of the 50,000 unique simulation conditions. The hypothesis would be partially substantiated if the models with augmented state spaces out-performed those without across the design space. The effect of geometry was less important but had to be controlled for.

The models could also be tested against the baseline guidance algorithms. Of the four baselines, only PN uses speed information in its algorithm; the rest are based solely on range and angle information. The baseline guidance algorithms were re-run on the smaller set of simulation conditions to facilitate fair comparisons.

Twenty-four models were trained for each combination of agent and state space. An important question had to be answered: Which subset of models would be used for the above comparisons? Two options were considered: Use the average over all 24 models, or use TOPSIS to select a smaller set of models from each group for analysis. The average might be more useful for comparing the expected performance of the model types against one another, while TOPSIS would give an optimistic view of how well a model of each type could perform. It was decided that the two model types would be compared at each design point using the average over the 50 geometries, representative of the expected performance from two network architectures. A single model from each would then be selected using TOPSIS on the 100,000 data points per model for comparison to the baselines, similar to the analysis processes used previously. The difference between the application of these techniques here versus previously is the additional dimensions from the design problem.

4.8.3 Non-Augmented State Space Model Results

Average capture rate versus design variable settings for the pursuer models trained without an augmented state space are shown in Figure 4.40. The data shown are the average over

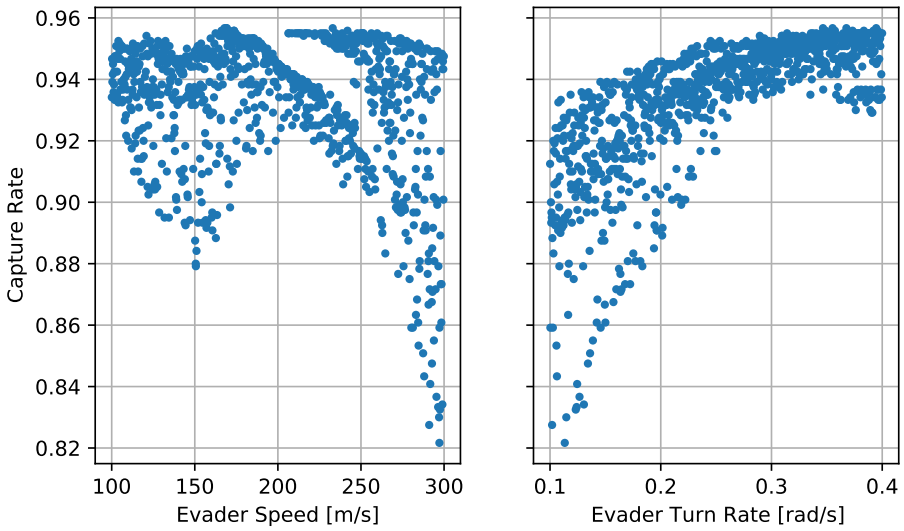
the 24 models and 50 geometries for each of the 1,000 design points in the two-dimensional LHS. Detailed analyses of these results were not warranted because the primary purpose of this step in the experiment was to compare the augmented and non-augmented state space models. To this end, only brief observations were made about the trends seen.

In general, the models appeared performed quite well. They were able to achieve consistently high capture rates against pure evasion, although they did not perform as well as PN, particularly for fast evaders with low turn rates. The precipitous drop in the metric as the evader speed approached the high end of the range when the evader used BE was unexpected. However, this may be explained by the nature of the beam evasion algorithm, which was designed to maximize the rate of change of the line of sight angle between the pursuer and evader. The best pursuit guidance counter to this might be to maintain a lead angle on the evader. However, that lead angle would be a function of the speed of the evader. Since the speed was unknown, the pursuer may have been unable to find a robust pursuit policy which could accommodate the range of evader speeds. It is also an average over the 24 models, leaving open the possibility that, while the majority fared poorly, there may have been at least one model which performed better.

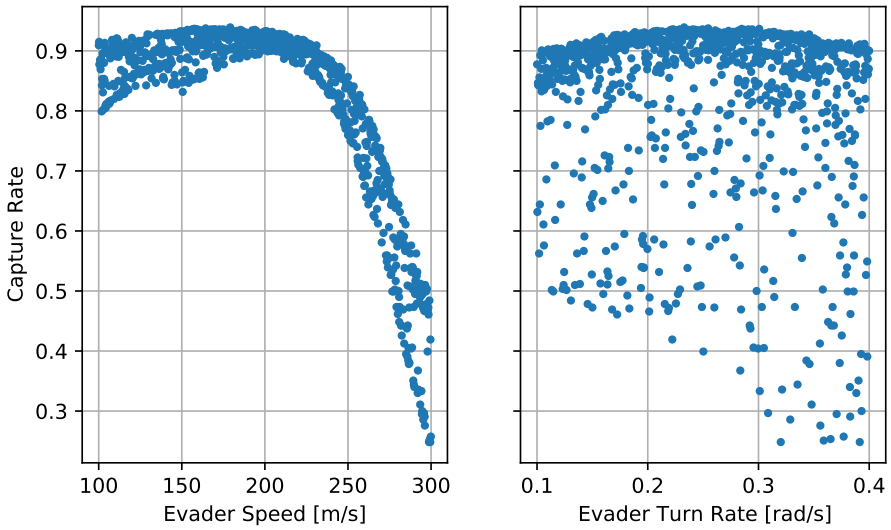
Results for the evaders are shown in Figure 4.41. The trends in the design space were very similar to those seen in the baseline cases. In general, the evader performed better against PP when it was faster and more maneuverable. However, these trends were not seen against PN, where the evaders appeared to have a harder time avoiding capture as they got faster. Notably, the maximum average capture rate against PN was less than 1.00, indicating at least one evader was able to learn a policy which could effectively evade PN.

4.8.4 Augmented State Space Models

Results for the pursuer models trained using the expanded state space are shown in Figure 4.42. These models appeared to perform better than the non-augmented state space models. The minimum average capture rate against PE was around 0.94, compared to 0.82

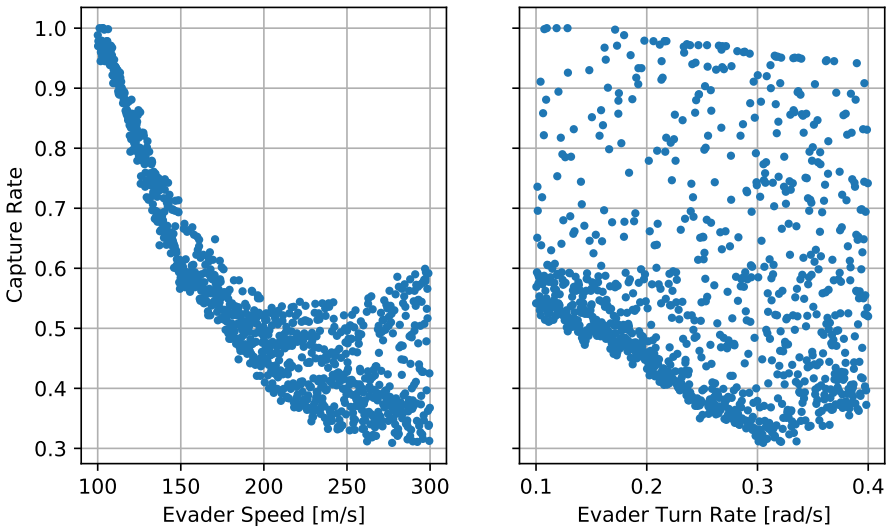


(a) Versus Pure Evasion

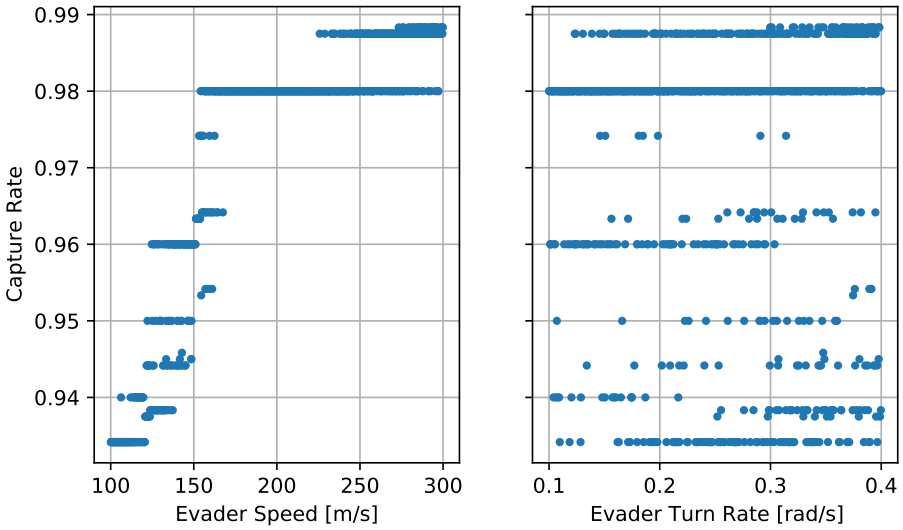


(b) Versus Beam Evasion

Figure 4.40: Capture rate versus design attributes for non-augmented pursuers against evaders using baseline guidance algorithms



(a) Versus Pure Pursuit



(b) Versus Proportional Navigation

Figure 4.41: Average performance of non-augmented evaders against pursuers using baseline guidance algorithms design attributes

for the non-augmented models. Notably, these minima both occurred when the evader was fastest and had the lowest turn rate. Against BE, the augmented models had an expected minimum capture rate of 0.6, compare to 0.3 for the non-augmented models. These minima occurred when the evader was fastest and had the highest turn rate.

Results for the evader models are shown in Figure 4.43. Any differences between the metrics for these models and those for the non-augmented models were difficult to identify visually. The minimum against PP was slightly lower at around 0.25, versus 0.30 for the non-augmented models. Metrics were very similar against PN.

4.8.5 Comparison Between State Spaces

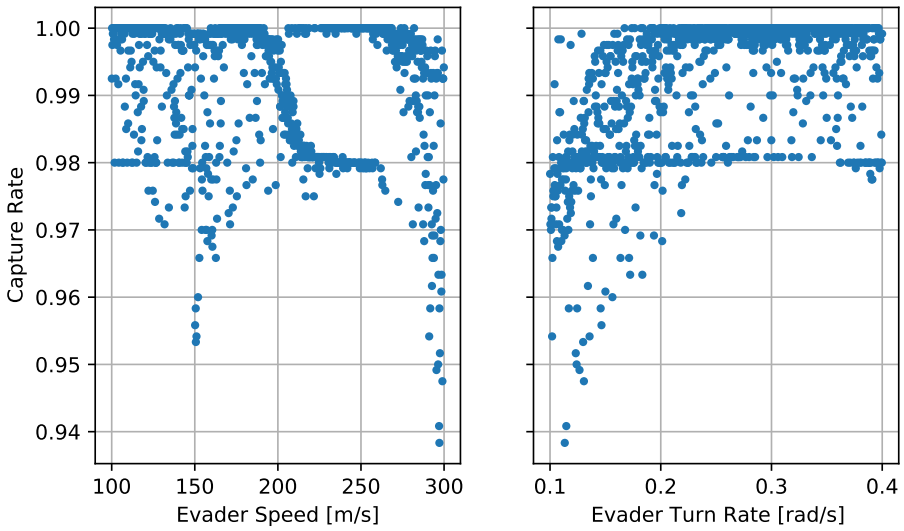
Pursuers

Differences in capture rate were calculated using the averaged data shown in Figures 4.40-4.43. Direct comparisons were possible because the two sets of models were tested on the same 1,000 design points and 50 geometries. The difference in capture rate $\Delta\bar{p}$ was calculated using (4.19), where $\bar{p}_{std}$ is the average capture rate for the models using the standard state space and $\bar{p}_{aug}$ is that for the models using the augmented state space.

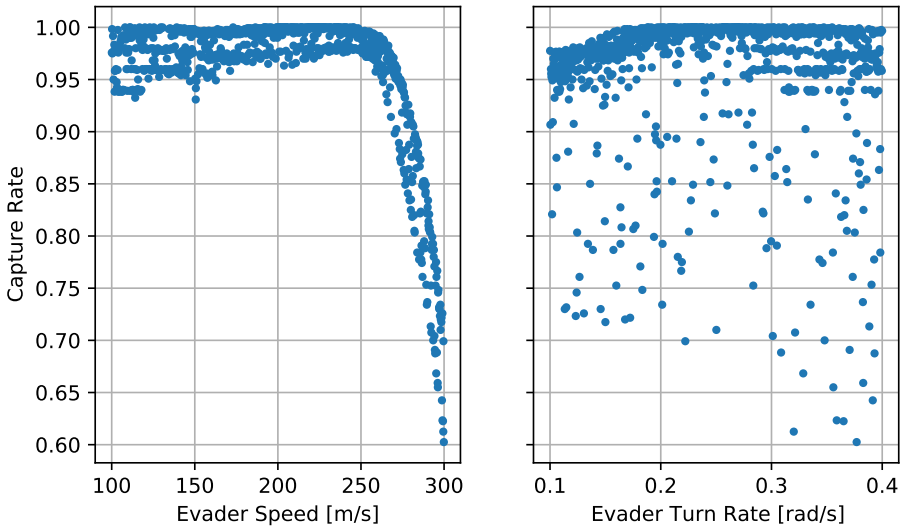
$$\Delta\bar{p} = \bar{p}_{std} - \bar{p}_{aug} \tag{4.19}$$

The first comparison was a non-parametric analysis of the capture rate difference. Histograms of $\Delta\bar{p}$ for pursuers are shown in Figure 4.44. The negative values indicated the expected capture rate for the augmented state space models was consistently higher than that for the non-augmented models. Distributions of the difference metric exhibited significant negative skew for both baseline evader guidance algorithms. Distribution metrics on the capture rate difference between augmented and non-augmented state space pursuer models are given in Table 4.27.

The capture rate differences for each design point are shown in Figure 4.45. The scatter

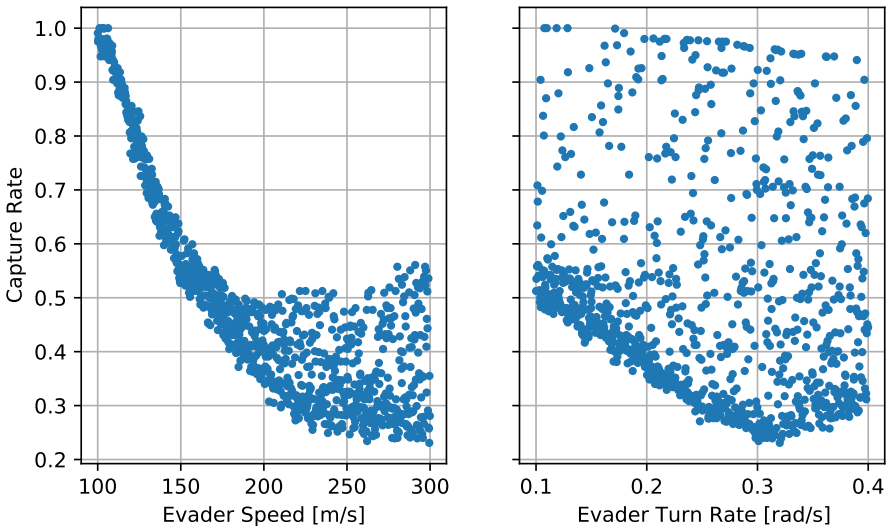


(a) Versus Pure Evasion

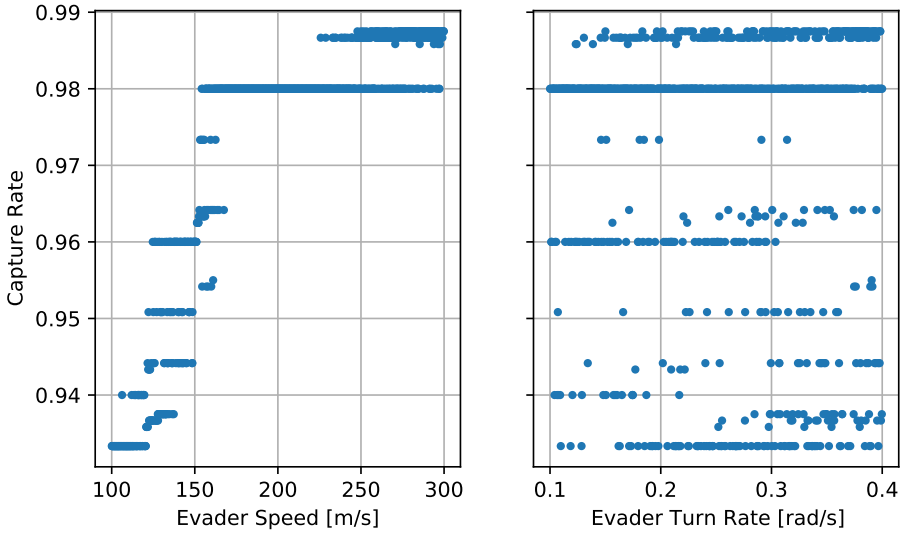


(b) Versus Beam Evasion

Figure 4.42: Average expanded state space pursuer model performance versus evaders using baseline guidance algorithms.

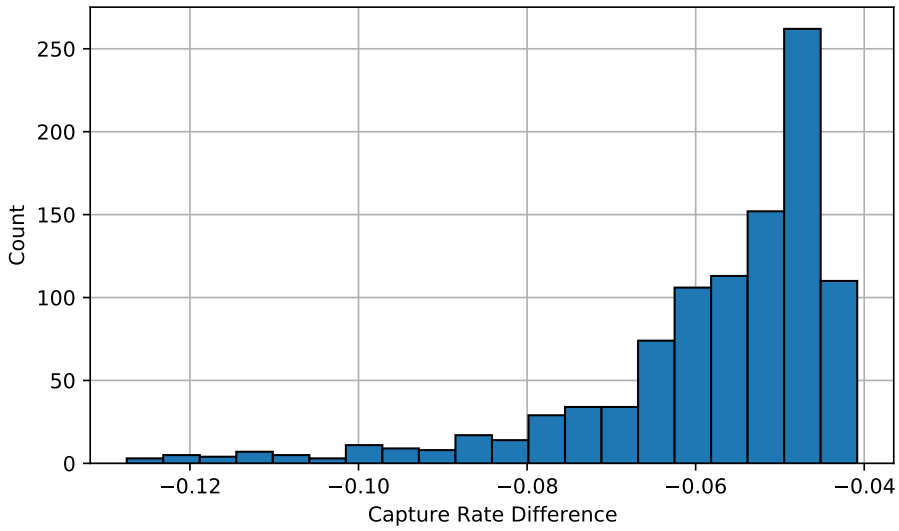


(a) Versus Pure Pursuit

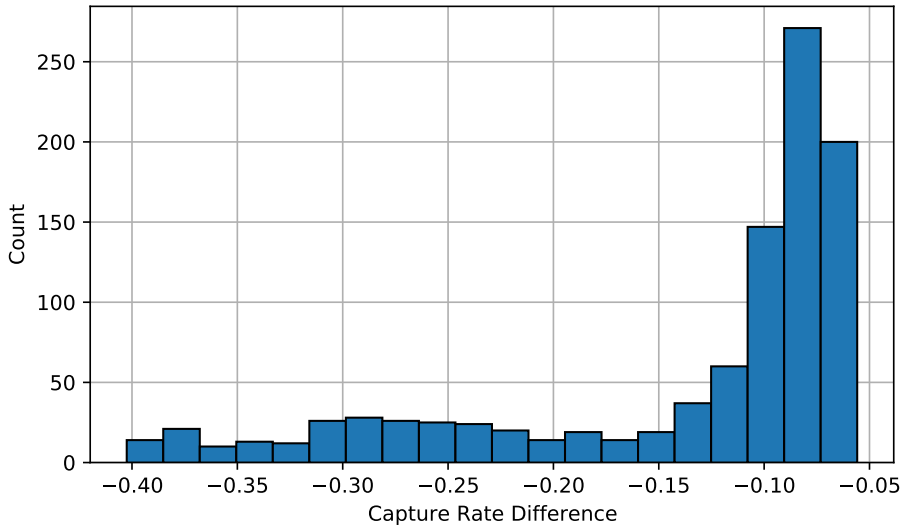


(b) Versus Proportional Navigation

Figure 4.43: Average expanded state space evaders model performance versus pursuers using baseline guidance algorithms.



(a) Versus PE

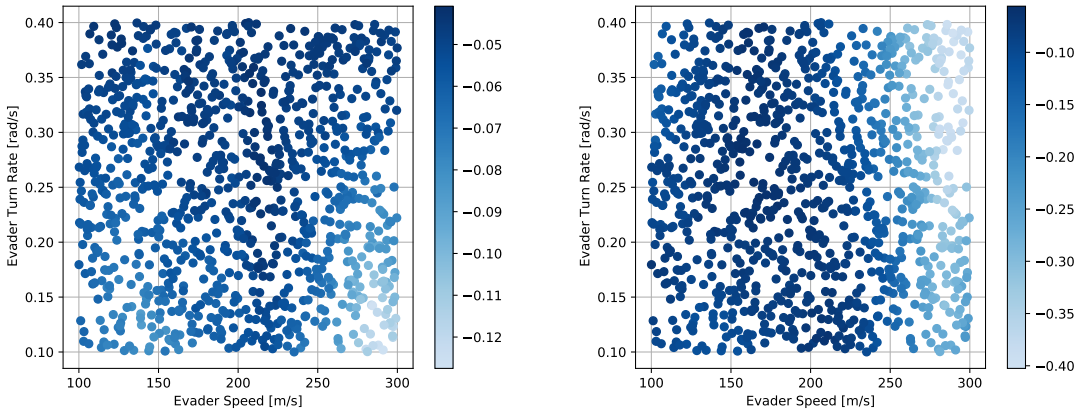


(b) Versus BE

Figure 4.44: Differences in capture rate between pursuer models trained with and without augmented state spaces. Negative values indicate augmented state space models had higher capture rate.

Table 4.27: Distribution metrics on capture rate difference for pursuers

Evader	Mean	Std. Dev.	Median
PE	-0.0580	0.0153	-0.0533
BE	-0.1381	0.0920	-0.0933



(a) Versus PE *(b) Versus BE*

Figure 4.45: Capture rate difference for pursuers over design space

plots showed the extreme differences in performance were confined to small regions of the design space, specifically those where the non-augmented models performed poorly. The magnitude of the differences in capture rate were at the low end over the majority of the design space. These observations lead to the conclusion that **the pursuers trained with an augmented state space were better able to mitigate the potential benefits to evaders offered by changes in design attributes.**

Evaders

Distributions of the capture rate difference metric for evaders are shown in Figure 4.46, with metrics reported in Table 4.28. The augmented state space models performed significantly better against PE and the non-augmented models, although there were a few cases where the latter out-performed the former. Against PN, however, the capture rates were practically

identical. The augmented models performed slightly better but not by a significant amount. This was attributed to effectiveness of PN and its use of the evader speed in the algorithm to calculate turning commands.

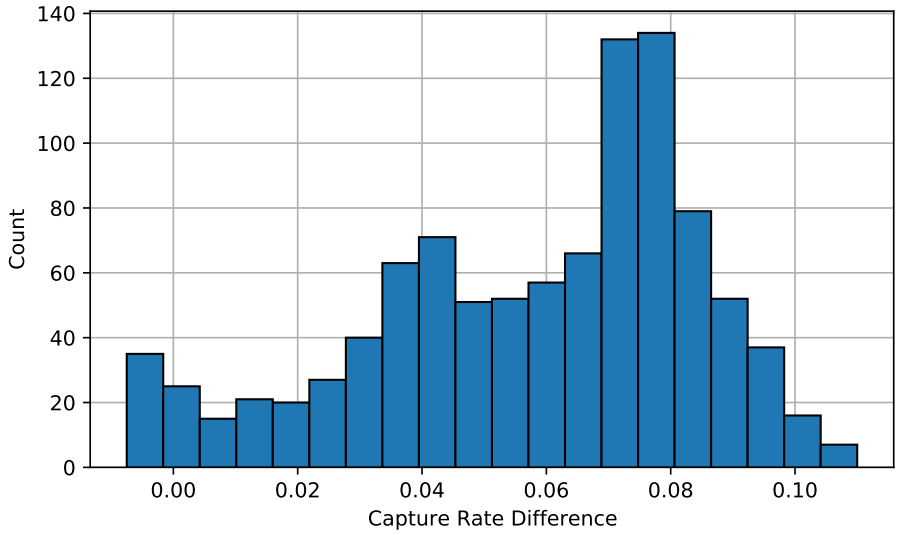
Table 4.28: Distribution metrics on capture rate difference for evaders

Pursuer	Mean	Std. Dev.	Median
PP	0.0579	0.0267	0.0658
PN	0.0002	0.0005	0.0000

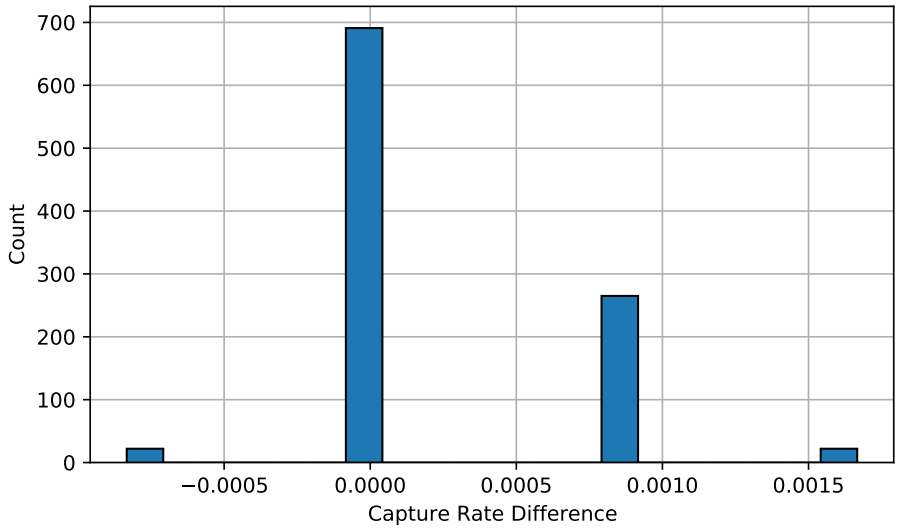
Differences in capture rate for the evaders against the baseline pursuers for each point in the design space are shown in Figure 4.47. The two models were captured at roughly identical rates at the low end of the evader speed range, independent of turn rate. The difference between the two was maximal at a speed around the middle of the design space around 220 m/s and at a higher, but not maximal turn rate around 0.35 rad/s. There was no difference between the two models against PN over the interior of the design space. Slight differences were seen at the edges of the speed dimension, and there were some trends visible along the turn rate dimension. However, these differences were not large enough in magnitude to warrant closer investigation. Overall, the results show **the models trained to control the evaders performed better across the design space with an augmented state space than without.**

4.8.6 Comparison to Baselines

Comparisons to the performance metrics of the baseline guidance algorithms over the design space were needed in order to test the state hypothesis. A single model was selected from each of the augmented model cases for this purpose. The non-augmented models were not included because it had been established that the augmented models were superior. The selection was performed using TOPSIS. The criteria used were the capture and end time metrics for each of the 1,000 design points tested, averaged over the 50 test geometries, for

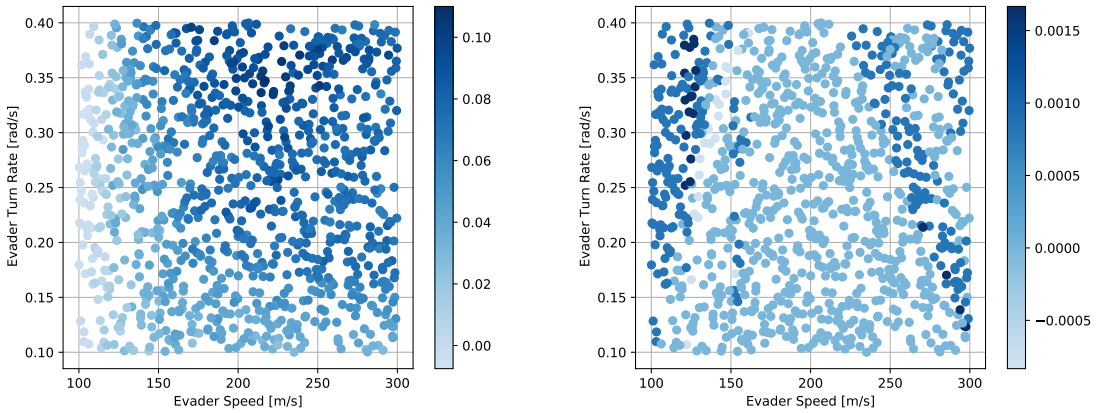


(a) Versus PP



(b) Versus PN

Figure 4.46: Differences in capture rate between evader models trained with and without augmented state spaces. Positive values indicate augmented state space models had lower capture rate.



(a) Versus PP *(b) Versus PN*

Figure 4.47: Capture rate difference for evaders over design space

each of the two possible opponent guidance algorithms. A total of 4,000 criteria was used for selection. The results are summarized in Table 4.29, where d_{best} and d_{worst} are the L^2 distances to the best and worst alternatives in each criterion, respectively. The similarity measure s_b was calculated using (4.16), where a value closer to 0 indicates closer proximity to the ideal solution.

Table 4.29: TOPSIS results for model down selection

Agent	**Index of Best**	d_{best}	d_{worst}	s_b
Pursuer	16	0.0714	0.4158	0.1465
Evader	6	0.8890	10.94	0.0752

Pursuers

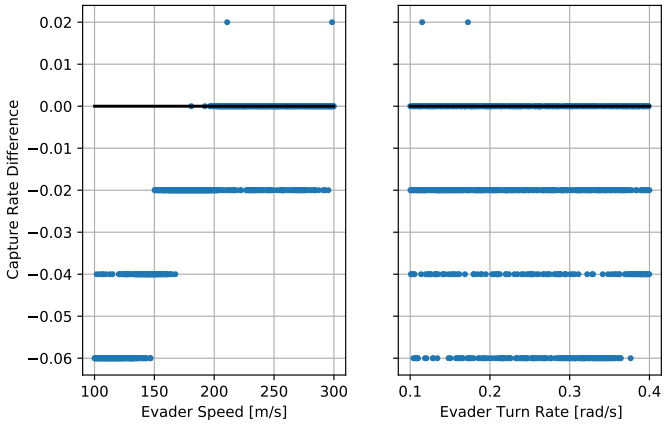
The capture and end time data for the pursuer model were collected for comparison to the baseline algorithms. The results for the three guidance models are given in Appendix B. Figure 4.48 shows the differences in the capture rate and end time metrics between the trained ANN and the PN guidance algorithm versus design variable settings. The differences were calculated using (4.20), where y_{Base} is the metric realized by the baseline PN

guidance and y_{ANN} that for the trained ANN. A positive (negative) value of Δy for capture rate would indicate evaders were captured at a higher (lower) rate when the pursuer used PN, relative to when the pursuer used the ANN. A positive (negative) value of Δy for end time would indicate more (less) time was needed to capture the evader when PN was used compared to the ANN. Pure pursuit was omitted from this analysis because PN was observed to perform significantly better, and therefore would provide a more optimistic baseline.

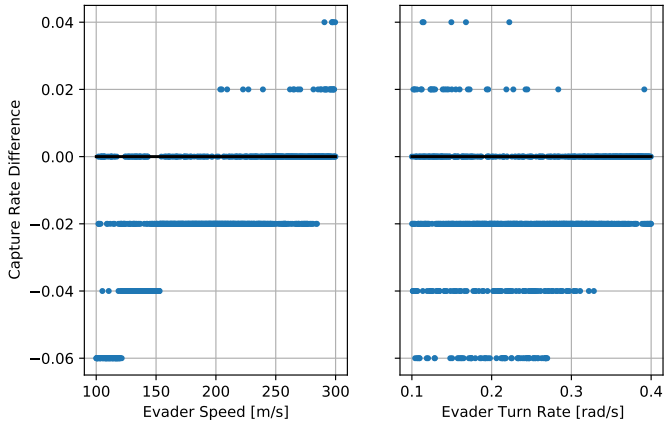
$$\Delta y = y_{Base} - y_{ANN} \tag{4.20}$$

The plots in Figure 4.48 show the difference in metrics Δy was negative for both capture and end time. This indicated the pursuer **captured evaders using either guidance algorithms more often when controlled by the ANN**, relative to when the pursuer was controlled by the baseline PN guidance algorithm. However, there were cases for which the PN guidance algorithm had a slightly higher capture rate than the ANN. These cases were concentrated in the high speed, low turn rate region of the design space. Further, **the pursuer took longer to capture the evader when controlled by the ANN** relative to the PN guidance algorithm.

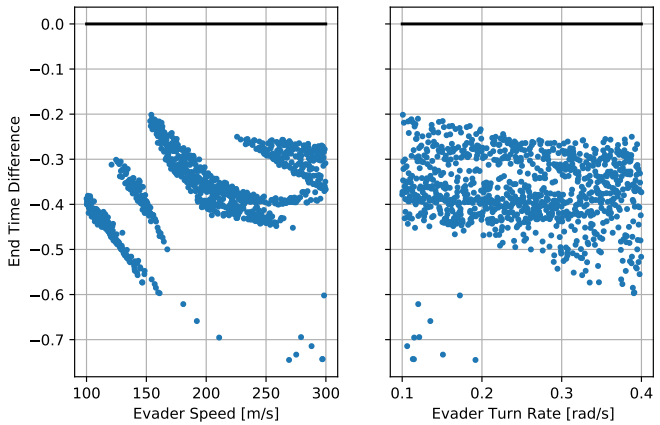
On average, over the entire design space, the ANN guidance increased capture rate by 0.021 against evaders using PE and 0.016 against BE relative to PN guidance. The average time taken by the ANN guidance to capture the evader was greater by 0.369s against evaders using PE and 0.287s against BE. These differences could be partially explained by the running reward mechanism (4.8) carrying a much lower weight in the overall performance index when compared to the terminal reward component (4.9). The trade-off between capture rate and end time favors increasing the former in this formulation.



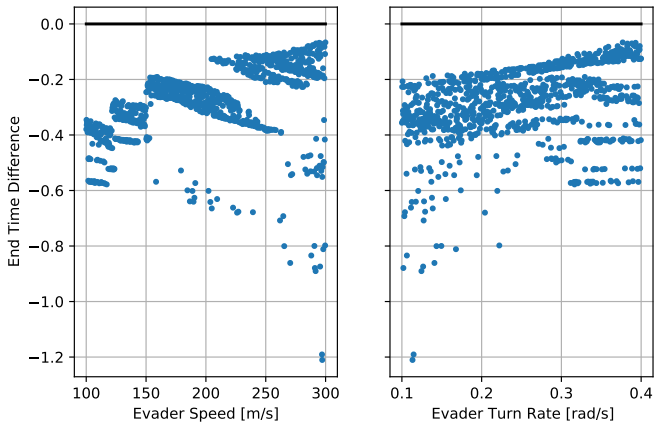
(a) Capture rate difference vs PE



(b) Capture rate difference vs BE



(c) End time difference vs PE



(d) End time difference vs BE

Figure 4.48: Differences in metrics between trained ANNs and PN guidance algorithm against both evader guidance algorithms versus design variable settings

Evaders

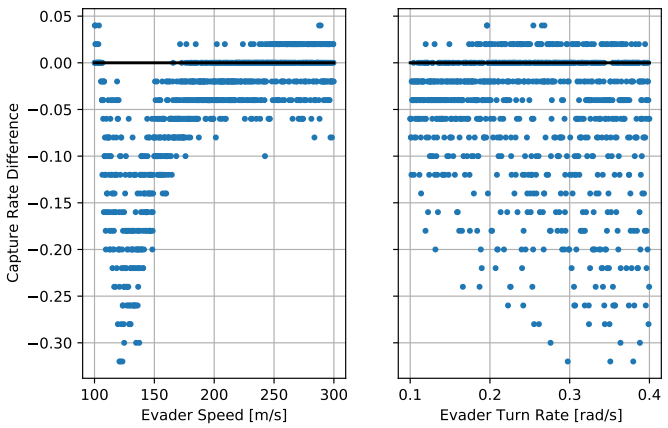
The differences in average capture rate and end time metrics between the ANN guidance and BE guidance are shown in Figure 4.49, again using (4.20) where y_{Base} corresponded to the metric for the baseline BE guidance algorithm. The PE guidance data were omitted from this analysis because it was observed that BE performed better in general and would, therefore, provide a more optimistic baseline. Data for all three guidance models are given in Appendix B.

The augmented models performed well in terms of capture rate against PN when compared to the baseline, as indicated by the positive values seen in Figure 4.49b. Th

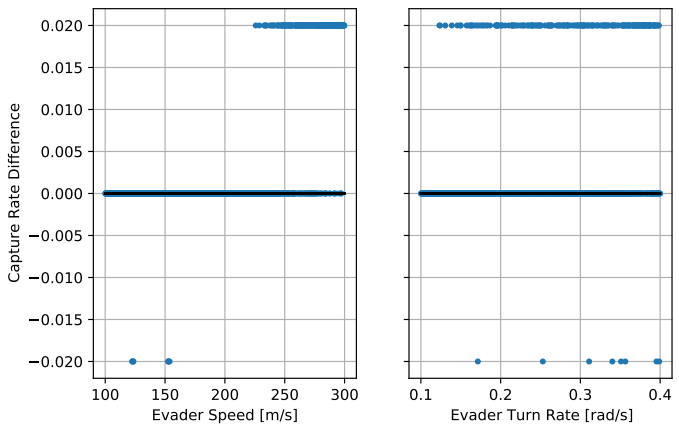
Comparison of Trajectories

The pursuer and evader models selected using TOPSIS were simulated against the baseline guidance algorithms on the five test geometries from Experiment 1 to compare their behaviors visually. Proportional navigation and beam evasion were selected as the baseline guidance algorithms for these tests, since those guidance algorithms were shown to be more effective for their respective agents under than the alternatives in most cases. Two values of evader speed and turn rate were selected for testing. The evader speed u_E was tested at 125 and 275 m/s, and the evader turn rate ω_E was tested at 0.15 and 0.35 rad/s. These values were selected because they were not on the very edge of the design space but were sufficiently different to illustrate how the behaviors of each agent would be affected. The minimum turn radius of the evader, corresponding to the minimum speed and maximum turn rate, was 357 m, while the highest turn radius was 1,833 m. A total of 60 trajectories was generated over the five geometries, four combinations of design variable settings, and three pairs of models. A single geometry was selected after a review in order to highlight the effects of changing the guidance algorithms. The trajectories of each pair are shown together for Geometry 0 in Figure 4.50.

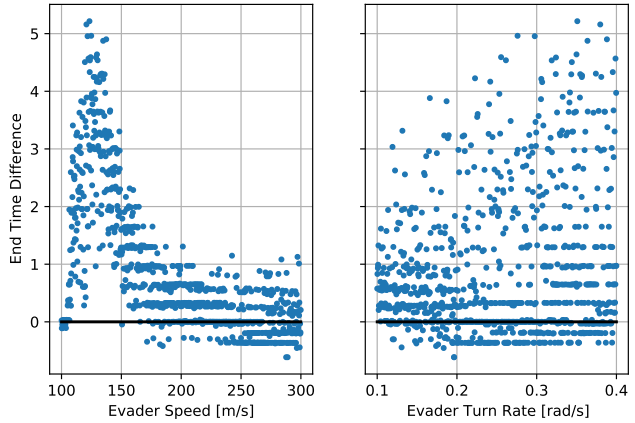
End time metrics for each combination of model pairing and design variable setting



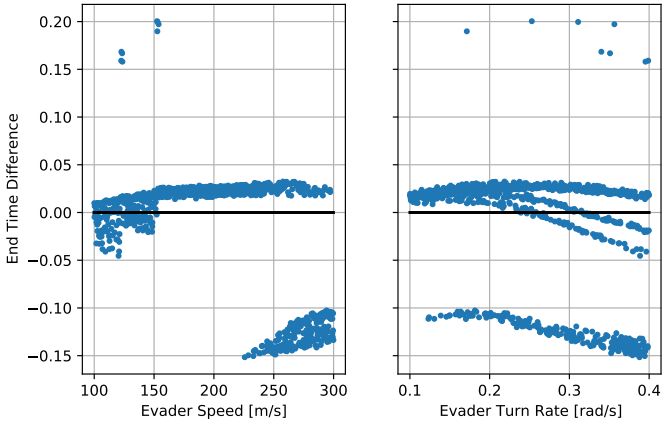
(a) Capture rate difference vs PP



(b) Capture rate difference vs PN

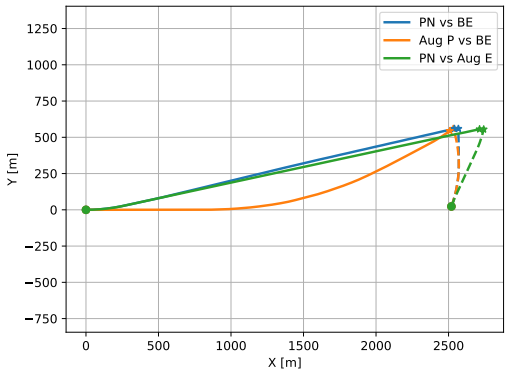


(c) End time difference vs PP

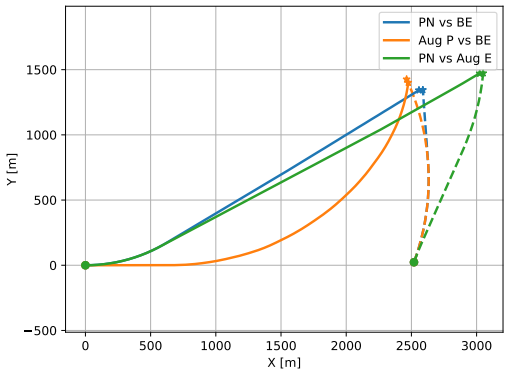


(d) End time difference vs PN

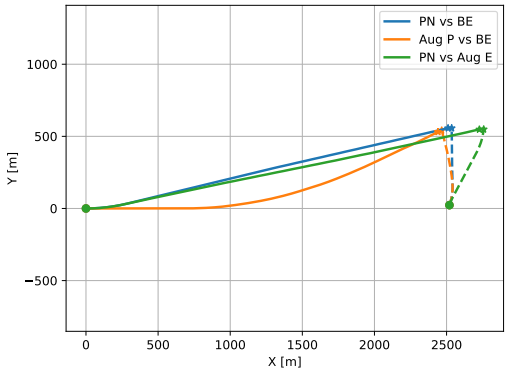
Figure 4.49: Comparison of evader metrics versus baseline pursuer guidance algorithms



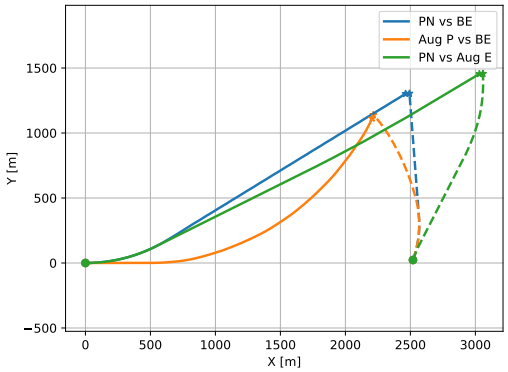
(a) $u_e = 125$, $\omega_E = 0.15$



(b) $u_e = 275$, $\omega_E = 0.15$



(c) $u_e = 125$, $\omega_E = 0.35$



(d) $u_e = 275$, $\omega_E = 0.35$

Figure 4.50: Test trajectories for combinations of model pairs and design variable settings on Geometry 0

are given in Table 4.30. The data provide several quantitative insights into the trajectories shown in Figure 4.50. For the baseline pair of models, the faster evaders were captured later than the slower ones, which was a fairly intuitive result. However, that the evaders with higher turn rates were captured faster than those with lower turn rates was non-intuitive. Examination of the blue lines in Figures 4.50b and 4.50c showed why this was the case. The tighter turn of the more-maneuverable evader put it in a position which was closer to the pursuer at the point where the 90° LOS angle from the former to the latter was achieved.

Table 4.30: End times for model pairings and design variable settings on geometry 0

$u_E\ [m/s]$	$\omega_E\ [rad/s]$	PN vs BE	Aug vs BE	PN vs Aug
125	0.15	4.35	4.38	4.63
125	0.35	4.30	4.24	4.64
275	0.15	4.87	5.23	5.68
275	0.35	4.70	4.41	5.65

The pursuer controlled by the ANN with an augmented state space performed slightly worse than one using PN against less-maneuverable evaders, but better against more-maneuverable ones. This was attributed to the distinct pursuit policy employed, where the pursuer maintained a straight course for several seconds before beginning to steer toward the maneuvering evader. This had the effect of causing the evader to continue its turn for longer, and therefore move towards the pursuer. The more-maneuverable evaders would turn more, resulting in a larger component of their velocity vector pointing in the direction of the pursuer. The pursuer could then follow a PN-like trajectory, maintaining side aspect into a successful capture. This strategy did not maximize the closure rate on the evader, but was evidently effective at capitalizing on the behavior of evader.

The evader controlled by the ANN with an augmented state space utilized an interesting, unanticipated strategy. It appeared to follow straight, PP-like trajectories which reduced – but did not minimize – the closure rate of the pursuer. It maintained this course until the

pursuer had come very close, at which point the evader initiated a hard turn across the path of the pursuer in a BE-like maneuver. This hybrid strategy allowed the evader to delay capture by up to 20% compared to the baseline BE algorithm while still attempting to maximize the LOS rate at a critical time to induce a miss. A pursuer using PP would likely not be able to respond to the sudden turn across its path and overshoot, allowing the evader to avoid capture. This was confirmed by visual inspection of trajectories where the ANN-controlled evader was simulated against a pursuer using PP.

4.8.7 Conclusion

The observations derived from inspection of trends across the design space for both pursuer and evaders model, trained using both the robust and augmented state space approaches, indicated **the models trained with an augmented state space were able to effectively capitalize on or mitigate any potential benefits afforded by changes in design attributes for the evader**. Further, examination of trajectories generated by ANNs with augmented state spaces against the baseline guidance algorithms showed the former were able to develop effective policies which capitalized on or mitigated both the design attribute information *and* the behavior of the opponent simultaneously.

4.9 Summary of Experimental Findings

The experiments presented in the preceding sections helped to substantiate the lower-level hypotheses concerning the fitness of the new elements in the proposed methodology. It was shown that ANNs trained using a standard RL algorithm could be used to produce models of behavior which effectively maximized expected performance and effectiveness compared to baseline algorithms. Further, it was shown that MARL with populations could be used in situations where baseline behaviors are not available. High throughput computing would be necessary to make such exploration and experimentation feasible, but the results shown here indicate the models could learn robust behaviors in a reasonable amount

Characteristic \ Alternative	1	2	3
Gap 3.1 State-Action Map	Mathematical Function	Decision Tree	Artificial Neural Network
Gap 3.2 Experimentation & Exploration	First Order Optimization	Zeroth Order Optimization	Reinforcement Learning
Gap 4 Engagement-Level Analyses	Multi-Objective	MDO	MARL
Gap 1 DSE	Partitioned	Robust	Augmented State Space

Figure 4.51: Revised morphological matrix of candidate solutions to the research objective. Selected alternatives are highlighted in green.

of time. Lastly, it was shown that the inclusion of design variable settings in the input to the ANNs could enable the models to develop policies which capitalized on or mitigate the potential benefits of the corresponding design attributes. This would allow for simultaneous exploration of the coupled technology and employment concept spaces in support of the overarching methodology for capability-based technology evaluation.

4.9.1 Statement of the Overarching Hypothesis

The morphological matrix of possible solutions to the identified gaps in meeting the research objective is shown in Figure 4.51. A new methodology for exploring employment concepts was to be formulated by selecting from these candidate solutions. Such a methodology would support and augment quantitative technology evaluation by filling a gap in the model construction step, as well as by helping to bridge gaps between design space exploration, employment concept exploration, and technology evaluation.

This chapter examined each step in the methodology and, where appropriate, identified relevant challenges. Each alternative shown in Figure 4.51 was considered against the

observations made in the previous chapters. Alternatives were selected based on those observations; the chosen alternatives are highlighted green in the figure. However, there was insufficient evidence to justify the selections outright. Corresponding hypotheses were then stated, to be substantiated via experimentation. Experiments were conducted on a representative problem in order to test the lower-level hypotheses regarding components of the overall method. The results of those experiments substantiated the hypotheses and demonstrated fitness of the selected morphology for meeting the research objective. The synthesis of the lower-level hypotheses into a proposed methodology, shown in Figure 4.52, formed the basis of the Overarching Hypothesis of this research:

Overarching Hypothesis

If artificial neural networks are trained to control interacting agents in an engagement-level agent-based model using multi-agent reinforcement learning and their state spaces are augmented using design variable settings then explorations of employment concepts in support of design space exploration will be possible

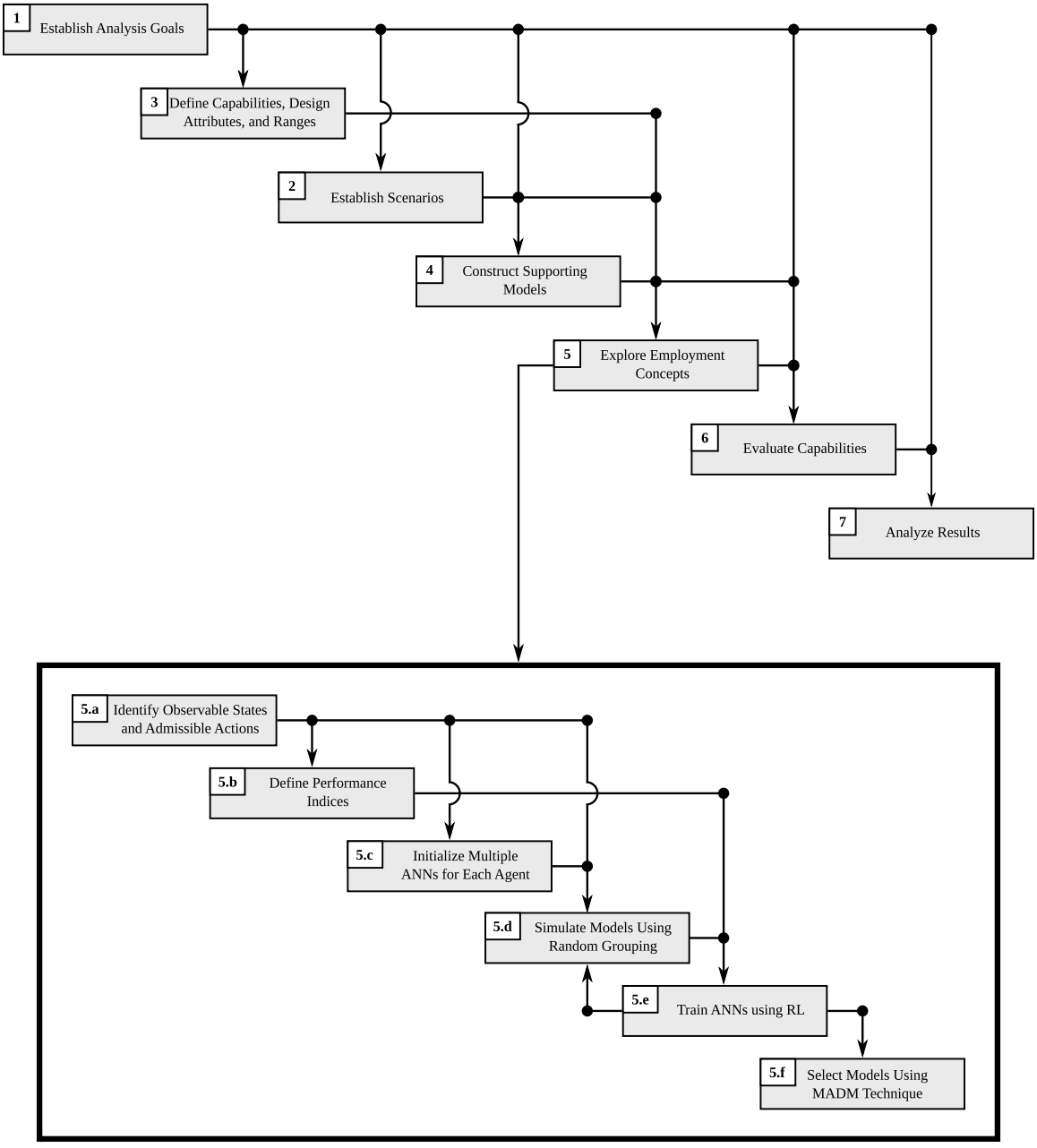


Figure 4.52: Completed methodology

CHAPTER 5
APPLICATION OF THE PROPOSED METHODOLOGY

"Scientific discoveries are made at the boundary of the descriptive power of the models that we use."

— Steve Mould

An application of the entire methodology was needed in order to test the overarching hypothesis. The goal of this final experiment was to demonstrate how the methodology could be used to enhance the analysis process. First, a suitable engagement-level problem had to be selected. The criteria used for selecting an experimental apparatus in the previous chapter had to be satisfied: The problem had to have multiple interacting entities and present a design problem where the behaviors of each entity would be affected by changes in design attributes.

The first criteria would be satisfied by nearly any problem relevant to the US DoD, so the search for a problem focused on the second. Recent acquisition programs provided a useful starting point to this end. The design of the F-35 could be revisited, with an emphasis on the engagement-level capabilities of the system and potential trade-offs between lethality and susceptibility. Various scenarios could be considered in for the multi-role aircraft, such as strike operations or offensive and defensive counter air. The Long Range Strike Bomber presented an interesting design problem with similar considerations as the F-35, but applied to a different type of engagement. The B-21 design focused on long-range, high-payload operations in contested environments [144]. A design problem could be formulated around the thrust-to-weight ratio and wing loading of the vehicle to explore the potential effects of e.g. engine technologies on effectiveness and performance. The SOCOM Armed Overwatch program is an effort to acquire a low-cost manned aircraft to replace the existing U-28A Draco in support of ground operations [140]. A design space

around maneuverability, susceptibility, and payload capacity could be explored to gain insights into how asset characteristics impact effectiveness in engagements.

Each of the aforementioned engagement-level problems would likely require a significant amount of effort to develop for the purpose of experimentation with employment concepts. Modeling even simple air-to-ground operations takes a considerable amount of time and effort to ensure all parts are interacting correctly. Consider a basic engagement involving a stationary target being defended by a stationary surface-to-air missile launcher against a bomber-type aircraft armed with air-to-ground. Five interacting models would have to be developed, along with guidance algorithms for the weapons. Appropriate values for launch and intercept conditions would have to be determined, either through estimation or additional experimentation. Sensor models would have to be developed if desired, along with appropriate signature data for each agent. Further, the necessary information may not be publicly available. These factors could adversely impact the validity of this experiment.

A balance had to be found between the simplicity of the pursuit-evasion scenario and a more substantial one for application of the complete methodology. Too complex and the experiment might become overburdened by questions of credibility to be useful. Too simple and the findings might not be significant enough to properly test the proposed methodology. A well-studied problem was identified based on these considerations: Air-to-air combat. Several examples of air-to-air maneuvering having been presented in this document, and interest in the exploration of air combat tactics for one-on-one engagements has been the subject of multiple studies over the past several decades [8, 154, 105]. However, only one of these studies considered a design problem associated with the engagement, leaving a large gap in the literature which could be addressed by this work.

5.1 The Air Combat Problem

Challenges in exploring air combat tactics have been touched on in previous sections of this work. The core question is: How do changes in design attributes for a fighter air-

craft impact performance and effectiveness in a one-on-one air combat engagement? Such engagements can be divided into two categories: Those for aircraft with similar characteristics and those for aircraft with dissimilar characteristics. Shaw defines dissimilar aircraft as those with performance characteristics which differ by more than 10% [123]. Primary characteristics for these comparisons are instantaneous and sustained turn rates, climb rate, linear acceleration, and minimum and maximum speeds.

5.1.1 Constraint Analysis

Performance characteristics can be mapped to two important descriptors for jet-powered aircraft: Thrust-to-weight ratio T/W and wing loading W/S, where T is the thrust produced by the vehicle, W is the weight of the vehicle, and S is the total wing or wetted area. Thrust varies with altitude and speed, and weight varies with the amount of fuel consumed and, to a lesser extent, weapon expenditure. For these reasons, the thrust term is usually specified using the sea-level value T_{SL} and a lapse parameter α is used to account for the effects of altitude and speed. Similarly, takeoff weight W_{TO} is in analysis and a lapse parameter β is used to account for fuel consumption or other effects. These two descriptors can then be used to estimate the performance limits of an aircraft using Mattingly's "master equation" (5.1), where q is the dynamic pressure, n is the load factor, C_{D_o} is the zero-lift drag coefficient, R captures the drag contributions from non-aerodynamic sources, V is the vehicle air speed, and E is the energy height of the system, given by (5.2) [81, 89].

$$\frac{T_{SL}}{W_{TO}} = \frac{\beta}{\alpha}\left[\frac{qS}{\beta W_{TO}}\left(K_1\left(\frac{n\beta}{q}\frac{W_{TO}}{S}\right)^2 + K_2\left(\frac{n\beta}{q}\frac{W_{TO}}{S}\right) + C_{D_o} + \frac{R}{qS}\right) + \frac{1}{V}\frac{dE}{dt}\right] \tag{5.1}$$

$$E = h + \frac{V^2}{2g_o} \tag{5.2}$$

The master equation can be used to generate constraint diagrams for a vehicle. This is

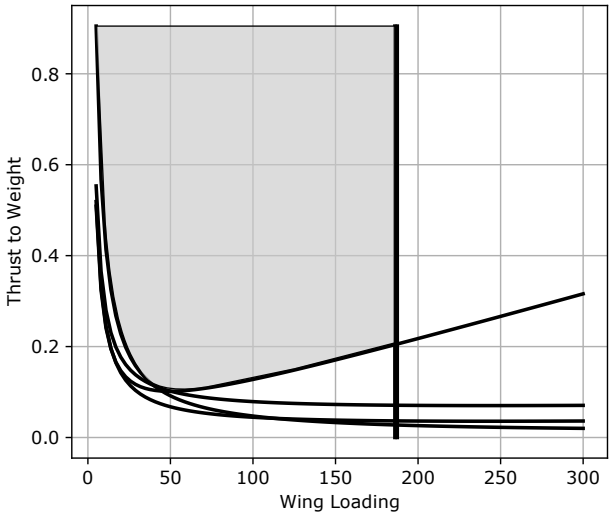


Figure 5.1: Notional constraint diagram. Feasible space is indicated by shaded region.

done by specifying the conditions which the design must satisfy in terms of the right-hand side of (5.1) and solving for T_{SL}/W_{TO} as a function of W_{TO}/S. Common examples of constraints used for analysis are takeoff, maximum linear acceleration, maximum sustained turn, and maximum rate climb. A notional constraint diagram is shown in Figure 5.1, where the feasible space is shaded grey. High wing loading can negatively impact performance, as (5.3) shows stall speed is proportional to the square root of wing loading, so the aircraft with a higher wing loading will have a higher minimum speed than one with a lower wing loading. Higher wing loading also necessitates higher cruise speeds by the same relation. These considerations indicate a low wing loading would be the desirable trait for a fighter.

$$W = L = \frac{1}{2}\rho S V^2 C_L \implies V_S = \sqrt{\frac{W}{S}\frac{2}{\rho C_{L_{max}}}} \tag{5.3}$$

Maximizing thrust-to-weight may be desirable for a fighter aircraft designed for one-on-one air combat engagements. This is because low T_{SL}/W_{TO} corresponds to low specific excess power P_S at a given speed, which is a measure of how well the aircraft can climb or accelerate [123]. This leads to a natural question: What values of thrust-to-weight ratio for a fighter aircraft in one-on-one engagements enable that system to meet the value

objective(s)? Research by Spearman indicated modern aircraft have thrust-to-weight ratios greater than 1.0 and combat wing loadings between 60 and 80 pounds per square foot [133]. His research also showed thrust-to-weight ratios had trended upward between the years 1945 and 1980, while combat wing loading had stayed relatively stable over the same time period. Analyses by the RAND Corporation indicated the thrust-to-weight ratios of military turbofan engines had been steadily increasing between the 1960s and early 2000s [161]. Additional research indicated the proportion of fighter aircraft structural weight attributed to lightweight materials, such as composites and titanium, had been trending upward between the years 1967 and 2008 [7]. While actual values were not readily available, these trends and some accompanying discussions indicate efforts have been made to both increase thrust-to-weight ratio and mitigate increases in wing loading for fighter aircraft.

5.1.2 Weapon Selection

The choice of weapons to be considered in this experiment was important, just as it would be in actual air combat. There are two primary methods for engaging: Missiles and guns. Missiles require sensors for tracking, targeting, and homing, and those sensors can use either infrared or radar, or both. Midcourse guidance for missiles typically relies on support from the launching aircraft, since the sensor systems on-board the missile are typically less capable than those on the aircraft and are only useful for terminal guidance. Missiles can do damage via two mechanisms: Kinetic energy transfer and/or detonation of an explosive charge. Kinetic kills are extremely difficult, since they require pinpoint accuracy against a high-speed, maneuvering target, not counting the even higher speed of the missile itself. Damage done by explosives decreases as the distance between the detonation point and the target increases, generally following an inverse-square law. The majority of the damage done to the target is from fragmentation of the missile body and integrated shrapnel. These projectiles are uncontrollable and their effect on the target will be random. They must hit a critical system in order to have a meaningful effect, and those systems can be shielded or

redundant in order to enhance survivability.

Guns are much simpler. They damage the enemy largely through kinetic energy transfer and, to a lesser extent, the mechanical destruction of critical systems. Sufficient damage from bullets can compromise aerodynamic performance and structural integrity. Ultimately, the likelihood of the system becoming inoperable increases as damage is accrued from bullet impacts. System redundancy and armor can improve survivability, but may come at a cost to performance by increasing vehicle weight. However, landing bullets on a maneuvering target can be extremely difficult. Bullets are aerodynamic bodies which move quickly through the air, but are unpowered and subject to the effects of gravity. Furthermore, mounted guns usually cannot be actuated to engage targets off-boresight. Precise maneuvering and positioning relative the target are necessary to utilize guns effectively in an air combat engagement.

The complexities of missile engagements make them unappealing for this experiment. Missiles can be fired from beyond visual range, reducing the importance of maneuvering and emphasizing detectability. Guns, by contrast, place heavy burdens on the maneuvering capabilities of the pilot and aircraft, and are much simpler to model. For these reasons, gun-only engagements were selected for this experiment. This also allowed for comparisons to be made between the results of the experiment and the available literature on these types of engagements.

5.1.3 Employment Concepts

As noted in the above discussion, maneuverability and positioning play important roles in analyzing gun-only air combat engagements. The goal of either aircraft is to maneuver into, and maintain as long as possible, a position where bullets can land on the enemy, deal damage, and down the aircraft. However, each is also attempting to avoid finding itself in said position relative to its adversary. This is a similar problem to the pursuit-evasion scenario, the main difference being the characteristics of the two systems and the

win condition.

The importance of the design problem becomes apparent through this lens. If one aircraft can maneuver significantly better than the other, either by turning tighter or being able to accelerate to a higher speed faster, then it may have an advantage in terms of being able to land and avoid hits with a gun. The primary question is: How can different turn and energy performance capabilities be leveraged to win the gun fight?

Shaw describes several tactics for gun-only engagements which are divided into two categories: Angles and energy [123]. An angles fight is characterized by pilots maneuvering in such as to place their enemy, and ideally their tail, within a small off-boresight angle. This can be achieved by executing tight, precise turns, but this might require reducing speed. Flying close to stall speed minimizes turn radius but can leave the system vulnerable if it is not able to rapidly gain energy and avoid a counter attack. This gives the impression that low wing loading and high specific excess power would be the desirable characteristics for an aircraft in these types of engagements.

As noted earlier, survivability is positively correlated with wing loading: More redundancies and armor increase weight. Further, the number, size, calibre of guns on the aircraft, along with the amount of ammunition carried, will factor into the takeoff weight. Bigger guns weigh more, but might be more capable of damaging enemies and therefore improve the chances of winning. These factors establish a trade-off between survivability and lethality, the likes of which was seen in the fights over the Pacific Ocean during World War II. The F4F and F6F were slow, heavily armored fighters which could pack a punch against the nimble but lightly armored A6Ms, and pilots of the former leveraged those dissimilarities in their tactics to stay in the fight.

5.2 Step 1: Establishing Analysis Goals

The purpose of this final experiment was to explore the design space of fighter aircraft in a one-vs-one, gun-only air combat engagement. The objective was to develop an under-

standing the trade space for potential technologies to enhance survivability and lethality, and how those technologies might impact system performance.

The primary MOE relevant to this problem was the likelihood of winning a gun-only engagement, and a value objective had to be established to reflect this. A win probability of 100% would be ideal, but such a threshold may be unreasonable. An actual threshold might have to consider the operating environment, the potential costs associated with losing, and other factors. An absence of these factors from the present context makes establishing a justifiable threshold very difficult. A value was taken from the available literature to formulate the objective: **The goal of this analysis was to explore how variations in design attributes might influence capacity for the system to achieve a win probability of at least 85%** [98].

5.2.1 Identifying a Simulation Environment

An M&S environment was needed to facilitate analysis. The Python environment developed for prior experiments could have been modified to accommodate the new problem, but some concerns were raised regarding the development of additional capabilities and validity on the more complex problem. The relatively simple pursuit-evasion scenario only required some basic mathematical manipulations and value tracking in order to simulate the necessary components. However, there are further considerations required for the gun engagement problem. In particular, it was desired that the tool used here be validated to some extent, so as to mitigate concerns regarding the implementation of the proposed methodology. Support was also needed for weapon simulation, which was not needed in the pursuit-evasion scenario, along with better sensing and tracking capabilities. The various needs imposed by this more-complex problem are visualized in Figure 5.2.

AFSIM was previously considered at the onset of the experimentation effort as a tool for M&S. It comes with ready-made modeling capabilities for 2D and 3D motion, sensing, weapon effects, communications, and more. These capabilities, along with its status as a

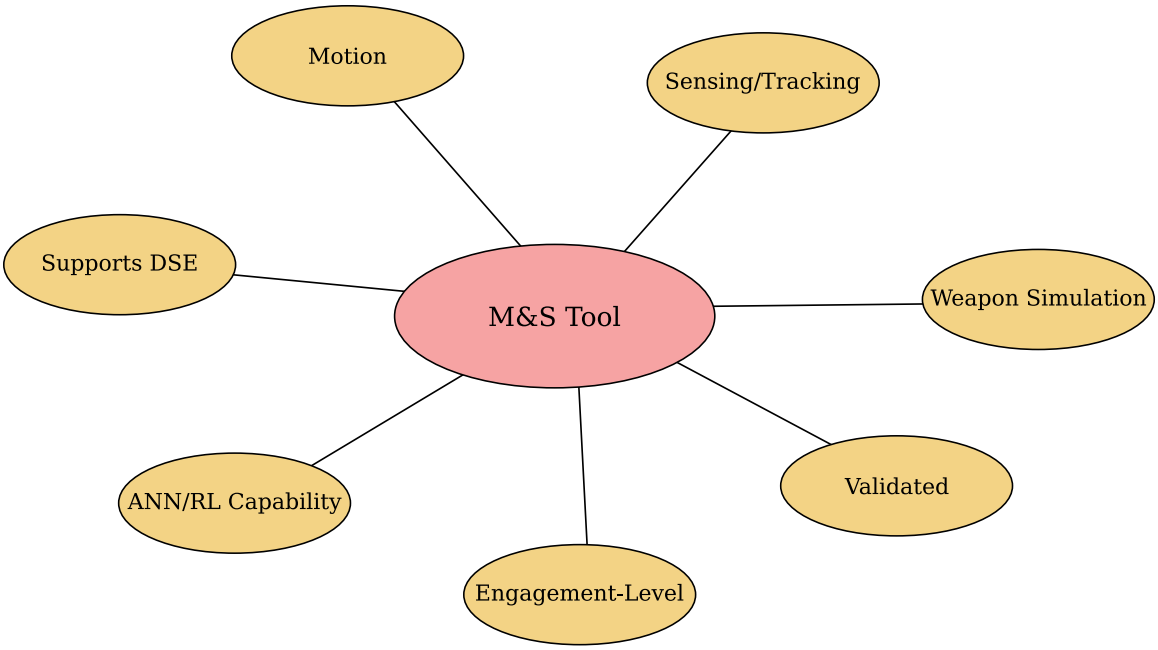


Figure 5.2: Necessary components of modeling tool to support experiment

standard tool within the defense M&S community, made it a more appealing environment for conducting this final experiment than relying on the custom-built Python environment. Leveraging a standard tool set would remove some potential concerns regarding validity of the model and implementation of the various algorithms employed to support the application of the methodology.

AFSIM did not possess an innate capability for evaluating or training ANNs using RL. However, AFSIM was developed to be extensible via user-built C++ plug-ins. Enabling training in AFSIM would have required a significant amount of "reinventing the wheel". The methods for autodifferentiation and gradient-based optimization implemented in freely available tools like PyTorch would have been cumbersome to translate and integrate into the AFSIM source code [102, 103]. Furthermore, AFSIM was primarily design as a tool for forward analysis with the purpose of outputting data for post-processing. Reinforcement learning requires a feedback loop, where the outputs are processed, the model updated, and the simulation rerun to produce more data. AFSIM has some limited capacity to perform these types of operations using file input/output methods to create self-modifying code, but

this was deemed too difficult to be feasible in the time available.

A plug-in was built and integrated into the source code to enable AFSIM simulations to load and evaluate ANNs. A capability was developed in Python to allow ANNs created in PyTorch to be converted into an AFSIM-compatible syntax. The AFSIM model was then constructed in a way which logged the state observation, action, reward, and simulation state (done/not done) information every time an ANN was evaluated. This information was written to a text file at the conclusion of the simulation, and a companion capability was developed in Python to read the data into a format for training using PyTorch.

5.3 Step 2: Defining the Design Space

Modern fighters enjoy a variety of sophisticated technologies, many of which are inaccessible by the public. For example, modern fighters have reduced radar cross sections, electronic warfare capabilities, extensive variations in weapon loadout, the ability to fly supersonic, advanced sensor fusion technologies, and provides a node for network-centric warfare [77]. There is little doubt of the effects of these technologies on the warfighter's ability to successfully and effectively complete missions, but many of them require highly detailed modeling efforts in order to be captured properly. However, a sufficient design problem could be constructed from much simpler principles than radar cross sections and stand-off missiles.

The design attributes considered here were informed by Shaw's analysis of the air combat problem. Turn rate, linear acceleration, minimum speed, maximum speed, damage output, susceptibility, weapon range, and off-boresight angle were considered. The first four are the traditional performance characteristics which can be used in constraint analysis. The last four are not directly used in constraint analysis but can inform selection of design points from within the feasible space.

5.3.1 Turn and Energy Performance

The basic aircraft characteristics considered in this experiment were derived from notional fourth-generation fighters and first principles. Combat speeds would not exceed Mach 1, or approximately 330 m/s, since doing so would require very high thrust, consume a significant amount of fuel, and not provide much advantage. A reasonable upper bound on combat speed would be 300 m/s based on this. Landing speed for an F-16 is reported around 83 m/s, so a reasonable lower bound on combat speed would be 100 m/s [37]. The range of 100-300 m/s defined the limits of capabilities to be explored. The baseline values were selected from within this range: 150-250 m/s.

Turn rate is a function of speed and load factor via (5.4), where g_o is gravitational acceleration and n is the non-dimensional load factor. The reported design load factor of a fourth-generation fighter aircraft is 9g. At 100 m/s, an aircraft executing a 9g turn would be capable of turning at a rate approximately 50 degrees per second. This drops to approximately 16 degrees per second at 300 m/s and 9g. However, a 9g turn cannot be sustained by a human pilot. A more realistic value might be 6g, corresponding to minimum and maximum turn rates of 11.08 and 33.25 degrees per second, respectively [104]. These values were used to establish the limits of turning capabilities used here. The lower limit was set to 10 $^\circ/s$, while the upper limit was set to 25 $^\circ/s$. Turn rate was assumed constant throughout the engagement, where less-extreme maneuvers would be made possible by rapidly switching between maximum effort turning and straight flight, identical to the implementation used in the pursuit-evasion experiments. This also necessitated a lower upper bound on turn rate, since a high-speed turn at 25 $^\circ/s$ might not be feasible.

$$\omega = \frac{g_o\sqrt{n^2-1}}{V} \tag{5.4}$$

Estimating linear acceleration capabilities for fourth-generation fighter aircraft was difficult. Specific excess power would have to be known, estimated, or assumed in order to

back out an estimate of the rate of change of kinetic energy, and an estimate of the mass of the vehicle would be needed to deduce the rate of change of speed. Thrust lapse as a function of air speed and altitude would also have to be factored in, requiring at least some information on engine characteristics. In lieu of such detailed information, a search through public data was used to inform the selection of upper and lower bounds on linear acceleration An AIAA undergraduate aircraft design challenge from 2005 provided upper and lower bounds on specific excess power for a "homeland defense interceptor" aircraft. The designs were to be capable of at least 200 ft/s specific excess power at Mach 0.9 at sea level, and as much as 400 ft/s at 15,000 ft [24]. The higher value was selected as the default specific excess power of the vehicle for simplicity and because more detailed engine analysis was not feasible. Specific excess power can be related to linear acceleration by (5.2), assuming the vehicle is not climbing so the rate of change of potential energy is zero. An estimate of the linear acceleration capabilities of the vehicle can be obtained using (5.5), where V_0 is the current vehicle speed and assuming the specific excess power is constant over one full second, i.e. $\Delta t = 1$. Further, assuming P_S is constant allows for dynamic calculation of the maximum linear acceleration the vehicle can produce, and this was the approach adopted here. The minimum value for P_S was set to 100 m/s, or 328.1 ft/s, and the maximum to 200 m/s, or 656.2 ft/s. These bounds were within the limits established by the AIAA design problem.

$$\Delta V \approx \sqrt{V_0^2 + 2g_o P_S \Delta t} - V_0 \tag{5.5}$$

The above calculations only applied to situations where the vehicle was attempting to *gain* speed. Different mechanisms would be at play in situations where the pilot wanted to lose speed, such as air brakes or simply letting off the throttle. In the latter case, acceleration would be driven by aerodynamic characteristics of the vehicle. Such attributes would be difficult, if not impossible to estimate for the purposes of this experiment. As a result, it was assumed that the vehicle could lose speed at a rate equal to the maximum rate at which

it could gain speed: 11 m/s^2.

Spatial Dimensionality of Engagements

Air combat occurs in three dimensions, and learning how to maneuver in that space is critical to fighter pilot training. The vertical dimension of maneuverability can also provide more alternatives for combat maneuvering. Avoiding disadvantages positions or achieving advantageous ones can be facilitated by climbing or diving, and energy management becomes increasingly relevant. The maximum altitude of the vehicle can also become a factor, and one must always be weary of crashing into the ground.

Modeling three-dimensional motion in a computer environment poses several challenges. Primarily, the equations of motion become more complex, with body angles factoring into the forward propagation in ways not relevant to two-dimensional motion. Energy considerations would require tying changes in altitude to changes in speed, and higher levels of fidelity would mandate considerations for air density and temperature as a function of altitude. This would impact thrust lapse and maximum load factor, and therefore practically all attributes of the system.

A decision was made to exclude the vertical component of maneuverability from this experiment because it introduced significant complexity in the model construction process. This was deemed necessary because of the added complexity from the design problem and associated increase in dimensionality of the state space. The additional realism from having three-dimensional maneuverability could unnecessarily complicate the analysis process; having a simpler 2D model allowed the experiment to focus on the interactions between the maneuvering and the design problems.

5.3.2 Damage and Susceptibility

Estimating the damage potential and susceptibility of an aircraft is very difficult for many reasons, some of which were discussed earlier. A reasonable estimate of gun damage

was used in a recent DARPA project exploring uses of RL in air combat simulation. The simulations calculated damage done with a gun as a function of the range between the two vehicles, specifically (5.6) [105]. This linearly decaying function roughly captures the effects of drag and dispersion on the ability of a gun to deliver kinetic energy to the target, and provided a reasonable baseline for this experiment in the absence of more information.

$$d_{gun} = \begin{cases} \frac{3000-r}{2500} & 500ft \leq r \leq 3000ft \\ 0 & \text{otherwise} \end{cases} \tag{5.6}$$

Either fighter in the simulation had a maximum "health" of 1. The value provided by (5.6) represented the damage done per second, and this value was accrued over the simulation. If the total damage received by a fighter exceeded 1 then that fighter was considered "killed" and to have lost the engagement. This allowed for the implementation of lethality and susceptibility parameters which scaled the damage done or received per second. A more lethal fighter would have its damage output scaled up, and a more survivable one would have its damage received scaled down. These were implemented as simple factors multiplying d_{gun} whenever damage was accumulated. A notional range was placed on each: Damage done could increase by as much as a factor of 1.5, and damage received could be decreased by as much as a factor of two.

Inherent to (5.6) is the maximum range and off-boresight angle at which the weapon is effective. The DARPA project used 3,000 ft or approximately 1,000 m, and 1 degree, respectively [105]. For design purposes, one might be interested in weapons which are effective at longer ranges or greater off-boresight angles. The maximum range was allowed to vary up to 4500 ft, or approximately 1500 m, based on the maximum range assessed by Siyu et al. [128]. As with the DARPA ACE project, no damage would be accrued at separations less than a threshold, in this case 100 m. Increasing the maximum had the effect of linearly increasing the amount of damage done as a function of range within 3000 ft. The off-boresight angle of ± 1 degree was found to be very restrictive and appeared to make

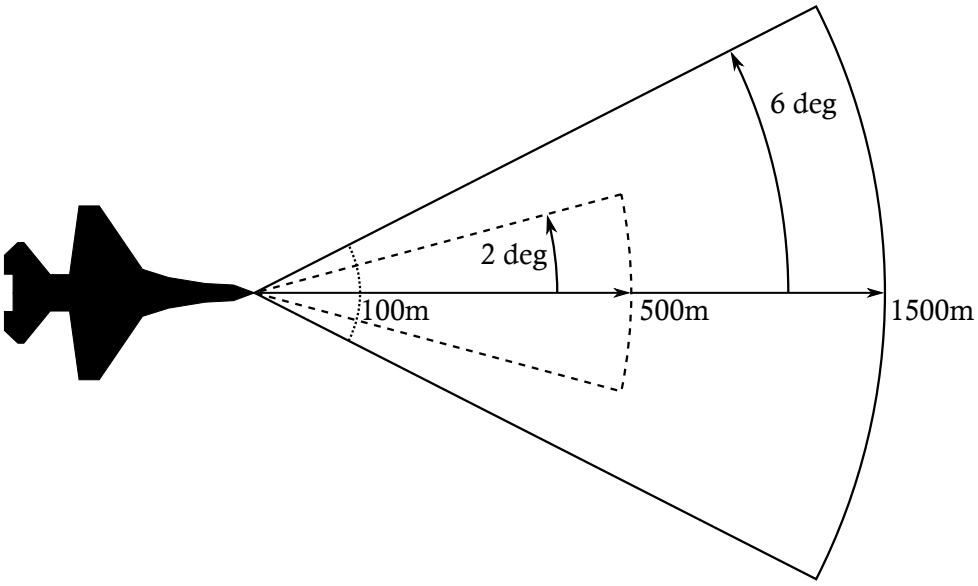


Figure 5.3: Weapon engagement zones for gun fight. Figure is not to scale.

learning more difficult, so the base value was increased to ± 2 degrees, and the maximum allowable value for the design problem was ± 6 degrees. These values were not based on any existing or future weapon systems, but were included in the design problem for the purpose of allowing explorations of possible advanced systems with better off-boresight capabilities in close quarter air combat. A lower limit of 500 m was imposed on gun range. This was done to capture the potential effects of sacrificing weapon capabilities for maneuverability. A smaller gun would weigh less and therefore reduce wing loading, increasing maneuverability. The limits on the weapon engagement zone are shown in Figure 5.3.

5.3.3 Summary

Eight design characteristics were selected for this experiment. The first four were relevant to the maneuvering problem inherent to gun-only air combat engagements; the latter incorporated considerations for lethality and susceptibility with regard to weapon employment. The characteristics and associated ranges are listed in Table 5.1, where the ranges were distilled from public sources and literature. Only one fighter will be subject to variations in design attributes; the other will use the default values listed in the table.

Table 5.1: Attributes and ranges for gun-only air combat engagement design space exploration

Attribute	Symbol	Low	High	Default	Units
Turn Rate	ω	10	25	15	deg/s
Minimum Speed	u_{min}	100	175	150	m/s
Maximum Speed	u_{max}	225	300	250	m/s
Excess Power	P_S	100	200	150	m/s
Damage Output Factor	λ_d	0.5	1.5	1	$n.d.$
Damage Taken Factor	λ_t	0.5	1.5	1	$n.d.$
Maximum Gun Range	r_g	500	1500	1000	m
Maximum Gun Angle	θ_g	2	6	4	deg

5.4 Step 3: Establishing the Scenario

Only a single scenario was considered for this experiment: Two fighter aircraft in a gun-only combat engagement. The aircraft were placed at notional altitude of 10,000 ft above sea level. Scenarios used for training the models were randomly generated in a manner similar to the previous experiments. The fighter whose design attributes were varied was initialized at a constant position and heading during all simulations. The other fighter was initialized at a random range and bearing relative to the first fighter, and given a random heading. The initial separation between varied from 500 meters to 5,000 meters. Initial angles were sampled between $\pm$180 degrees. Both fighters had their initial speeds randomly sampled from their respective ranges of possible speeds. Both fighters also had their initial damage value set to a random number between 0 and 1, which allowed the models to experience the full range of possible values from the onset of training. All sampling was done with uniform distributions. Training simulations were allowed to run for a maximum of 30 seconds simulated time.

The initial damage factor for each platform was randomly initialized between 0 and 1 for each training simulation. This was done to allow the models to learn how the damage

factor might influence their behavior or that of their opponent, without having to rely on the autocurriculum.

5.5 Step 4: Constructing Supporting Models

Only a few models were needed to support this experiment. Motion was enabled through the use of the standard kinematic mover type in AFSIM. This mover type enables smooth two-dimensional motion and control using route-following commands. The two fighters used the same mover with capabilities in excess of what would be used during simulation and training. Movers were forced to updated at a rate of at least 100 Hz to ensure smooth calculation of positions and other dependent metrics. Maneuvers were limited by the route generation algorithm used to convert ANN outputs into control signals.

The damage accrual model described previously was implemented in AFSIM using built-in observer methods to check the relative positions between fighters. Accrual was checked every time a mover was updated, and the damage done or received was scaled by the update time interval to account for this. Individual bullets did not have be modeled because of this, significantly reducing run times.

5.6 Step 5: Exploring Employment Concepts

5.6.1 Step 5.a: Identifying States and Actions

The same basic state observations used in previous experiments – range, relative bearing, and relative heading of the adversary, plus normalized simulation time – were carried over this one, although some new states were added to accommodate the more complex scenario model. The damage factor of both the self and adversary were included in the state space, along with the speeds of both the self and adversary. These had to be included because damage factor was used to determine the end condition and speed was a controllable state. The eight design attributes were also included. The observable states, excluding design

attributes, are listed in Table 5.2. Several of these states were linearly transformed before being passed to the network to facilitate learning.

Table 5.2: Observable states for air combat engagement

State	Symbol	Units
Range	r	km
Relative Bearing	ω	rad
Relative Heading	θ	rad
Own Damage Taken	d_s	$n.d.$
Adversary Damage Taken	d_a	$n.d.$
Own Speed	u_s	m/s
Adversary Speed	u_a	m/s
Simulation Progress	$\tilde{t}$	$n.d.$

Each fighter could control its heading in the same manner as the pursuit-evasion scenario. Turns were executed at maximum effort. Speed could be controlled through maximum accelerations, either positive or negative. The magnitude of acceleration was determined by solving (5.5) using the instantaneous platform speed. Speed and heading were could not be controlled independently, i.e. the platform could not simultaneously accelerate and turn. These choices were based on the implementation by Austin et al. and their remarks about rapid maneuver switching [8].

5.6.2 Step 5.b: Defining Performance

Terminal Rewards

The most relevant MOE was whether the engagement was won or lost, or ended with neither agent accruing sufficient damage to win. The terminal reward mechanism (5.7) was designed to reflect this. Agents received large rewards for winning the engagement,

moderate penalties for losing, and small penalties for ending in a draw.

$$S = \begin{cases} +50 & \text{if } d_a \geq 1 \wedge d_s < 1 \\ -30 & \text{if } d_s \geq 1 \wedge d_a < 1 \\ -10 & \text{otherwise} \end{cases} \tag{5.7}$$

Running Rewards

The running reward mechanism was composed of three parts. The first part, given by (5.8), was adapted from the work by Austin et al. for automated combat maneuvering with rotorcraft [8]. It consists of a range component which decays exponentially. This component is highest when the target is at the desired range $r_{g,opt}$, which was defined as $1/3$ the maximum range of the gun. The second component takes the relative angles of the platforms into consideration. It is maximal when both $\omega = 0$ and $\theta = 0$, which indicates the target is directly in front of the agent and facing directly away from it. Further, it is minimal when the target is directly behind the agent and facing directly at it. This formulation helps guide the agent towards more desirable behaviors: Keep the target in the weapon cone while staying out of its weapon cone.

$$L_{Austin} = \exp\left(-2(r - r_{g,opt})^2\right)\left(1 - \frac{|\omega| + |\theta|}{\pi}\right) \tag{5.8}$$

Austin et al. developed a reward mechanism which allowed for considerations of agent speed. However, this mechanism was difficult to implement because it required experimentation with four coefficients governing the reward curve as a function of speed. The premise of the curve was to reward the agent for maintaining a speed near an ideal value, and decaying that reward as the agent deviated from it [8]. The intent of the reward curve was captured using a simple quadratic relationship (5.9). This curve penalizes the agent for flying at extreme speeds, and favors minor deviations from the midpoint between the

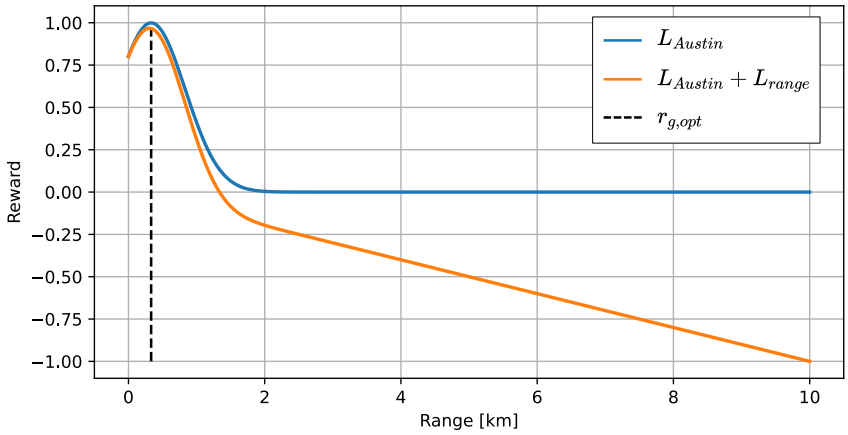


Figure 5.4: Comparison of base and modified range reward mechanisms

maximum and minimum allowable speeds.

$$L_{speed} = \frac{-1}{10}\left(-1 + 2\frac{u_s - u_{min,s}}{u_{max,s} - u_{min,s}}\right)^4 \tag{5.9}$$

The last component of the reward mechanism was designed to encourage the agents to close the gap between one another and engage. It was found that the exponential decay in (5.8) quickly drove the reward to zero because of the magnitude of the argument $2(r - r_{g,opt})^2$. However, scaling this component down would result in an excessively flat curve which would not accurately reflect the desire to keep the target in the weapon cone. The range penalty had a small impact at low values, leaving the exponential curve large intact, while adding an increasingly significant penalty at greater separations. A comparison of the basic L_{Austin} reward and the combined reward is shown in Figure 5.4

$$L_{range} = \frac{-1}{10}r \tag{5.10}$$

The three components were combined into the single running reward mechanism (5.11). The entire reward was shifted down so that it never took a positive value. This was done to encourage the models to resolve the engagement as quickly as possible, since they would be constantly accruing penalties as the simulation unfolded over time. Allowing the value

to cross zero would also make comparison more difficult because positive and negative values could sum to zero, possibly obfuscating the effects of both.

$$L = -1 + L_{Austin} + L_{speed} + L_{range} \tag{5.11}$$

5.6.3 Step 5.c: Initializing ANNs

The population approach to MARL was used here, with twenty-four models were initialized for both platforms. Each ANN used the architecture from Experiment 3, where the first hidden layer was split into two channels – one for the observable states from the environment and the other for the design attribute settings. The networks were identical to those used in Experiment 3, shown in Figure 4.39.

5.6.4 Steps 5.d and 5.d: Simulation and Training

Simulations were executed one at a time for each pair of models. AFSIM has the ability to run batches of simulations automatically, but the manual approach allowed for greater control of the process since the number of samples produced by each simulation could not be knowable. However, the design variable settings used by each pair of models during one episode were the same across all simulations in that episode. This was done to balance diversity of experiences in both spaces. The high number of episodes practically guaranteed the models would experience points across the entire design space. Simulating multiple engagements using a single set of design attributes was expected to allow for better representation of the potential effects those attributes would have on the engagement.

The models were trained using PPO with hyperparameters given in Table 5.3. A minimum of 2,000 samples were generated for training during each episode. Each simulation would produce at most 600 samples, so each pairing experienced at least four simulations per episode for data generation. A total of 50,000 training episodes were performed.

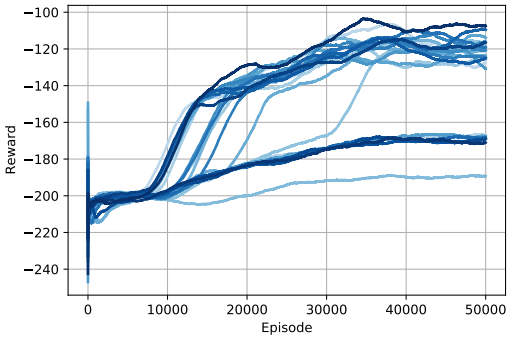
Table 5.3: Model hyperparameters

Hyperparameter	Value
Learning rate	3e-4
Batch size	1000
Epochs	5
Sample size	2000
Entropy	1e-4

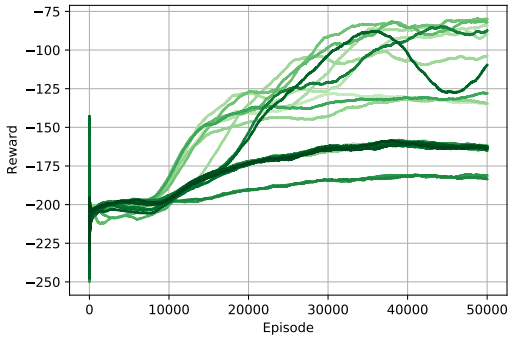
Training Results

The trends in average reward, win rate, and loss rate versus episode of training for both groups of models are shown in Figure 5.5. The models spent a significant amount of time exploring without discovering effective policies, as indicated by the flat trends up to episode 10,000. Most models began to improve significantly after that point. However, some did not improve significantly over the full duration of training. This further reinforced the importance of running multiple models.

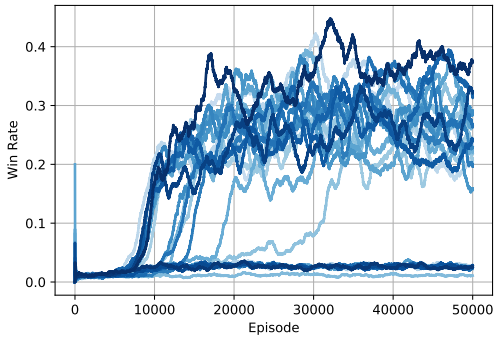
Figure 5.5 shows the standard fighters were able to achieve higher reward and win rate metrics than the designed ones. This was attributed to the range of design variables explored, which included degradations in performance characteristics. Figures 5.5e and 5.5f provide some insights on this phenomenon. The designed fighters were primarily grouped at an average loss rate between 0.15 and 0.20, with a smaller group closer to 0.25 and a single model at just 0.10. There was a dense group of standard fighters with a loss rate above 0.25 and another around 0.10, with a few models dispersed between them. Models with high loss rates would have inflated the win rate of opponent models, so the better performance of the standard fighters could be reasonably explained by the group of poor-performing models for the designed fighter with high loss rates.



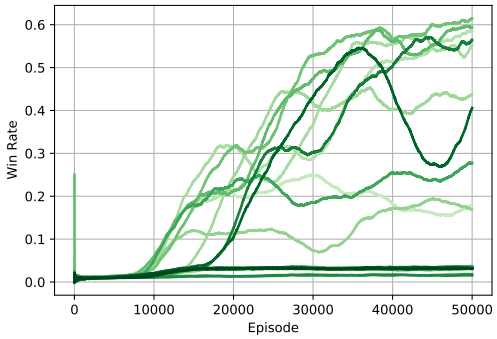
(a) Designed fighter rewards



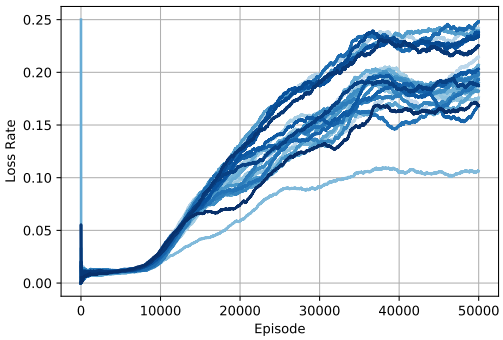
(b) Standard fighter rewards



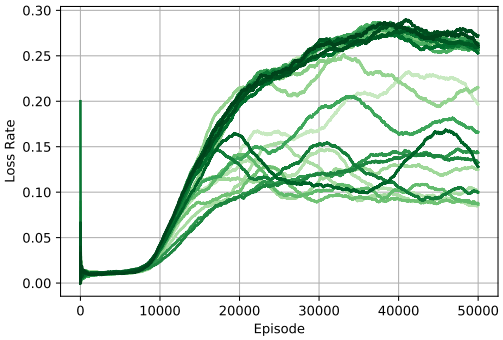
(c) Designed fighter wins



(d) Standard fighter wins



(e) Designed fighter losses



(f) Standard fighter losses

Figure 5.5: Trends in metrics for each group of models versus training episode

5.6.5 Step 5.f: Model Selection

TOPSIS was used to determine which model from each group would be carried forward through the remainder of the methodology, where the criteria were the reward, win rate, and loss rate averaged over the last 1,000 episodes of training. These metrics were chosen because they were readily available and were expected to provide a reasonable estimate of performance against all opponents and over the design space because of the random sampling employed during training. The index, metrics, and similarities of the chosen models are reported in Table 5.4. The results show the top model for the standard fighter – model 8 – was the best in all three metrics among all 24 models created, as indicated by the similarity measure of 0. The top-performing model for the designed fighter was significantly closer to the ideal than the second or third place model, although not as decisively. The two top-performing models from each group were selected to be carried forward for analysis and evaluation of the design space based on these observations and the previous experience with TOPSIS results.

Table 5.4: TOPSIS results for trained models

Group		**1st**	**2nd**	**3rd**
Design	Index	23	21	17
	Reward	-108.6	-110.0	-112.5
	Win Rate	0.3754	0.3219	0.3129
	Loss Rate	0.1589	0.1878	0.1992
	Similarity	0.1244	0.2276	0.2591
Standard	Reward	-81.97	-83.49	-83.61
	Index	8	2	9
	Win Rate	0.6180	0.5866	0.5784
	Loss Rate	0.0811	0.0876	0.0933
	Similarity	0.0	0.0471	0.0628

5.7 Step 6: Evaluating the Design Space

Each of the four pairings of top-performing models was simulated on 4,000 design points sampled using an LHS over the 8-dimensional space. Each design point was evaluated using 100 simulations on pre-generated initial conditions sampled using an LHS from the aforementioned test scenario distributions. The win/loss condition, damage done, and total reward for each of the simulations were recorded for analysis.

5.8 Step 7: Analyzing Results

Analysis was broken into two parts. The first was a statistical analysis of the design space based on the data produced by the four model pairings. The intent of the first analysis was to gain insights into how likely the designed fighter could be *expected* to win as its design attributes varied. Analysis of this data could inform decisions about regions of the design space were further effort and analyses should be directed.

The second part of the analysis was more focused and derived from the first. Insights into how performance varied over the design space was used to select individual cases for inspection and comparison. This would allow more detailed analysis into *how* the design attributes enabled or hindered the fighter over the course of the engagement.

5.8.1 Win Probability as a Function of Design Attributes

Model 23 vs Model 8

The probabilities of the designed fighter controlled by Model 23 winning the engagement against the standard fighter controlled by Model 8 are shown as a function of the design variable settings in Figure 5.6 for the designed fighter. The black lines indicate the centered average over a $\pm 5\%$ variation in that parameter without controlling for the effects of the others.

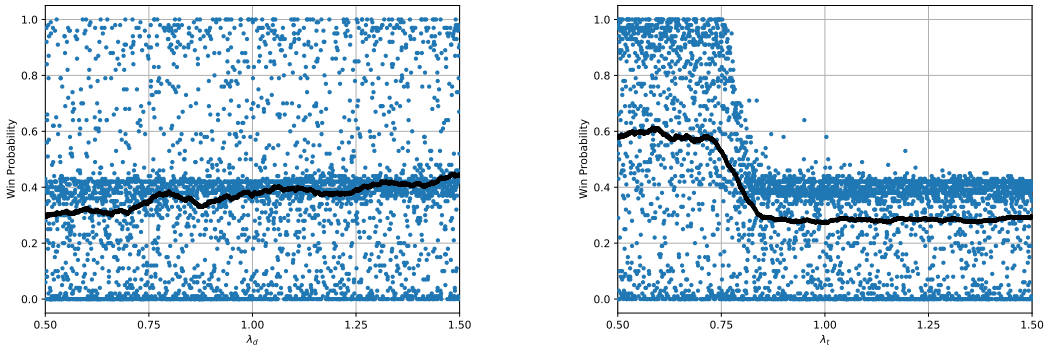
There were visible trends over the design space. The two most notable effects were

those attributed to susceptibility λ_t and turn rate ω. High susceptibility and low turn rate were both associated with significant decreases in expected performance for this pair of models. Low turn rates appeared to be highly detrimental to effectiveness, as the average win probability dropped below 0.10 when $\omega < 13^\circ/s$. Isolation of the cases where $\omega < 15^\circ/s$ showed the probability of winning was partially driven by the weapon parameters λ_d, r_g, and θ_g. Win probability was positively correlated with each of those parameters, indicating a less-maneuverable fighter could still win if its weapon capabilities were sufficiently above those of its opponent. These trends were visible in Figure 5.6 but were more pronounced in cases where the fighter was less maneuverable.

The susceptibility parameter $\lambda_t < 0.75$ was associated with a relatively high win probability, with several cases having a probability at or approaching 1. However, a precipitous drop in expected win probability is seen in Figure 5.6b beginning at 0.75 and end roughly at 0.85, where the expected value settles around 40%. There were a few cases for which the expected win probability was greater than 45%, despite having $\lambda_t > 0.85$. Isolation of these cases showed they were associated with high gun angles and ranges, generally higher values of λ_d, and turn rates greater than $15^\circ/s$. This suggested the negative trait of being a more susceptible could be partially offset by increases in lethality parameters, as well as some additional maneuverability. These observations, taken alongside the trends in effectiveness with respect to the turn rate parameter, indicated winning the gun-only engagement was predominately decided by survivability and maneuverability.

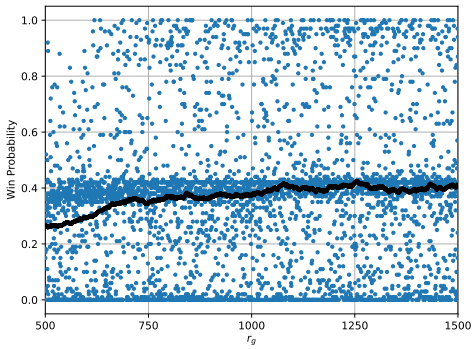
Model 23 vs Model 2

Results of the design space exploration where the designed fighter was controlled by Model 23 and the standard fighter by Model 2 are shown in Figure 5.7. Some of the trends for this pair of models were similar to those seen previously, such as those with respect lethality λ_d and turn rate ω. However, the susceptibility parameter λ_t had a substantially smaller impact on win probability than when the standard fighter was controlled by Model 8.

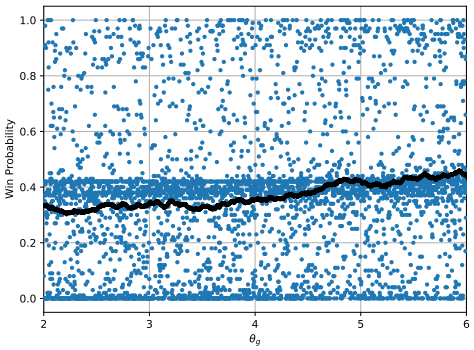


(a) Versus Lethality

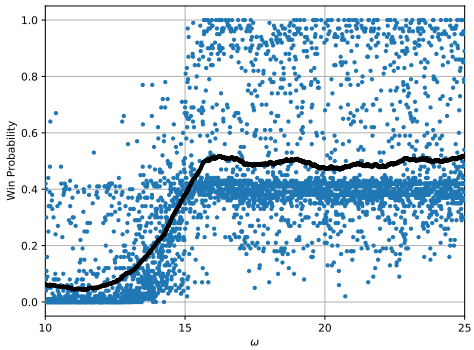
(b) Versus Susceptibility



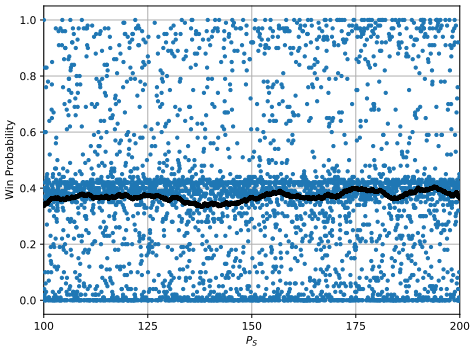
(c) Versus Gun Range



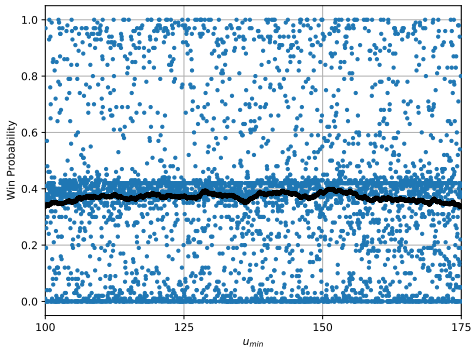
(d) Versus Gun Angle



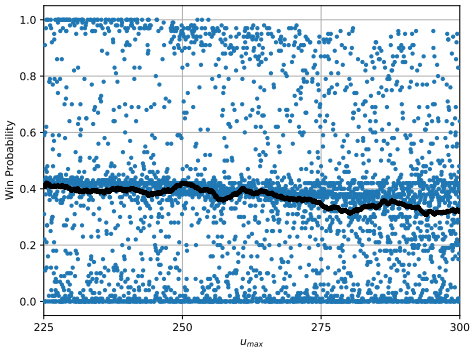
(e) Versus Turn Rate



(f) Versus Excess Power



(g) Versus Minimum Speed



(h) Versus Maximum Speed

Figure 5.6: Win probability for designed fighter versus design variable settings. Designed fighter was controlled by Model 23, standard fighter by Model 8

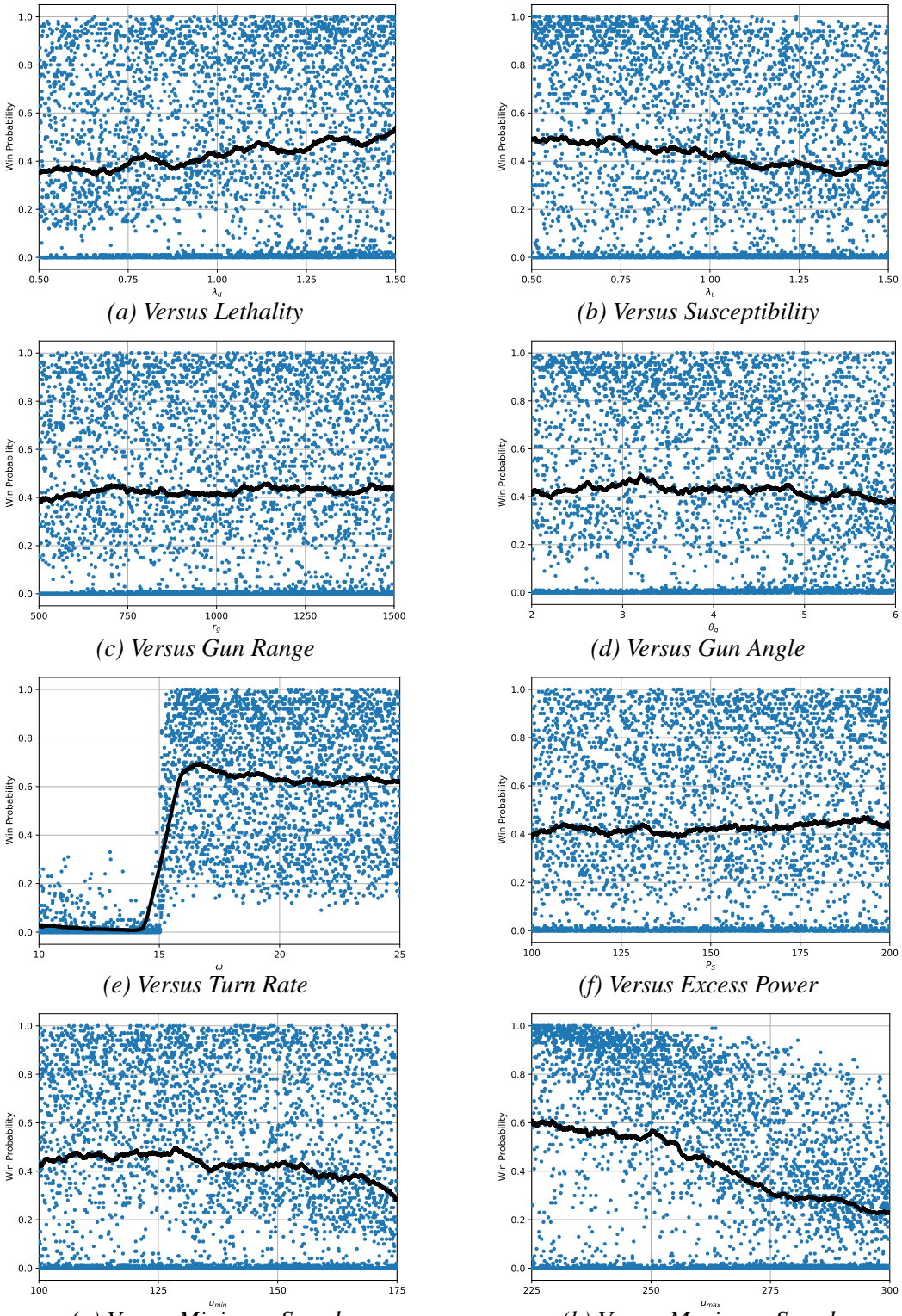
The effect of turn rate was a much more dramatic step response here than in the case of Model 23 versus Model 8. This was largely a result of the consistently low win probability for cases where $\omega < 15^\circ/s$ seen here. The primary implication was that even a minor disadvantage in turn rate was sufficient to nearly guarantee a loss of the engagement. A turn rate advantage, however, did provide a higher expected win probability in this case than in the previous one. Designed fighter Model 23 had a prominent band of win probabilities in the neighborhood of 0.40 against standard fighter Model 8 when $\omega > 15^\circ/s$; win probability was more dispersed against standard fighter Model 2 over the same range. Further inspection of cases where $\omega > 15^\circ/s$ showed the designed fighter largely relied on higher lethality and lower susceptibility to win, which was similar to the earlier findings. In the cases where $\omega < 15^\circ/s$, win probability was strongly driven by lethality λ_d, gun range r_g, and gun angle θ_g. However, susceptibility λ_t had little effect.

There was a much more significant effect attributed to the speed limits, with higher values adversely impacting performance. The effect of a higher minimum speed was largely expected, as discussed previously. However, it was not immediately clear why maximum speed had such an adverse impact, especially since the two parameters had extremely small effects against Model 8.

Comparison of Trends

Comparisons of trends in win probability across the eight-dimensional design space produced by each of the four pairs of models are shown in Figure 5.8. The trends generally agreed with one another but were not identical, particularly with respect to susceptibility λ_t, gun characteristics r_g and θ_g, and the speed limits u_{min} and u_{max}. The data where the designed fighter was controlled by Model 21 are shown in Appendix B.

Although the trends are visually similar, there are several distinctions between them which highlight the importance of conducting explorations of employment concepts to support design space exploration. For example, the apparent severity of decreasing turn rate



(a) Versus Lethality

(b) Versus Susceptibility

(c) Versus Gun Range

(d) Versus Gun Angle

(e) Versus Turn Rate

(f) Versus Excess Power

(g) Versus Minimum Speed

(h) Versus Maximum Speed

Figure 5.7: Win probability for designed fighter versus design variable settings. Designed fighter was controlled by Model 23, standard fighter by Model 2

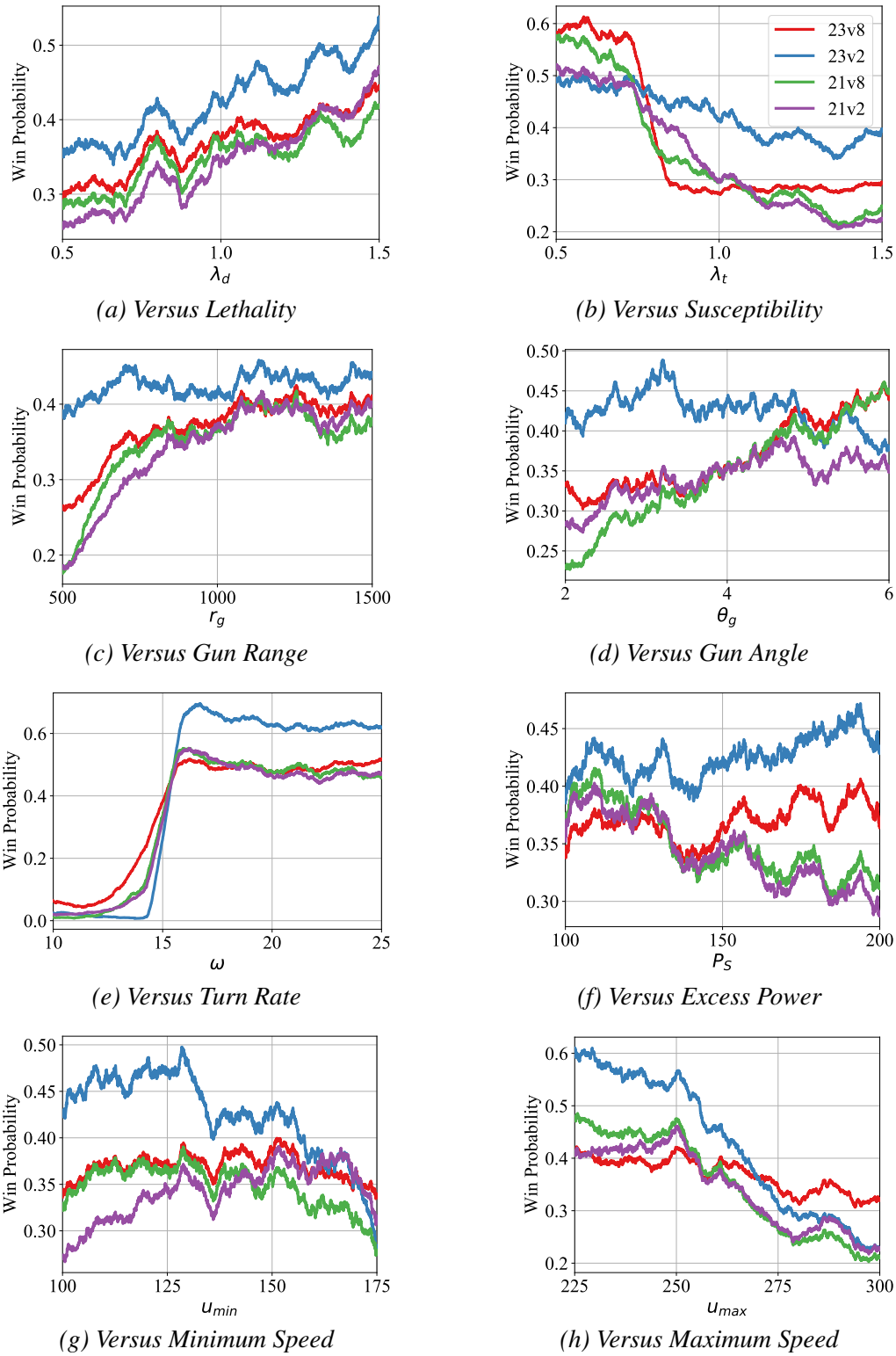
would differ depending on the choice of model for each fighter because the rate at which win probability decreased as turn rate decreased varied between the four cases. The effect of the susceptibility parameter might also be misrepresented: It could be a precipitous drop at values above 0.75, or a more gradual decrease across the entire range. The data produced by this experiment showed how the nature of the trends with respect to such variables could depend upon the choice of behavior models used to support the analysis effort.

The similarities seen between the trends in Figure 5.5 highlight certain consistencies with respect to the design attributes. Win probability was always positively correlated with lethality λ_d and, while was not surprising, the consistency across the four cases reinforced the significance of this parameter, independent of the others. Win probability was also negatively correlated with susceptibility λ_t, although the trends were less consistent. Observation of these trends could allow for a more confident statement of the impact a design attribute has on system effectiveness and subsequent effects on design decisions.

It should be noted that similarities in observed trends did not mean the same values were obtained across the space. This is most apparent in Figure 5.8a, where all four lines showed similar trends but the line corresponding to Model 23 vs Model 2 was shifted upward compared to the other three. The difference between the highest and lower centered average of win probability was as high as 10% over the range of values for λ_d. There were greater variations seen for other parameters, such as gun range r_g and minimum speed u_{min}. It would be imprudent to claim any one estimate was more accurate than the other, but the existence of such variation in values further substantiates the need for these types of explorations in the analysis process.

5.8.2 Inspection of Trajectories

The second part of the analysis was to inspect trajectories generated by the combinations of models for specific combinations of design attributes. The purpose was to enable examination of how different decision-making processes effected different outcomes, and to



(a) Versus Lethality

(b) Versus Susceptibility

(c) Versus Gun Range

(d) Versus Gun Angle

(e) Versus Turn Rate

(f) Versus Excess Power

(g) Versus Minimum Speed

(h) Versus Maximum Speed

Figure 5.8: Averaged univariate trends in win probability versus design variable settings for four pairs of models

compare the models more directly than would have been possible using data over the entire design space.

Selection of Design Attributes

Settings for the design attributes had to be selected for generation and comparison of trajectories. Prior analyses indicated there were two archetypes for the designed fighter which performed well: Highly-durable designs, and highly-maneuverable ones. These two archetypes reflected a trade-off between system characteristics, as more durable systems would have to incorporate protections against the opponent's weapon systems, such as sub-system redundancy. This would increase the weight of the system, consequently increasing wing loading and reducing maneuverability. This trade-off was seen in the design of the Grumman F4F and F6F fighters, which were notably durable aircraft and could perform well against the more agile Mitsubishi A6M by leveraging their lethality and survivability in concert with distinctive tactics.

At the other end of the spectrum are the designs which focus on maneuverability to achieve positional advantage. The general notion is that a weapon is only meaningful if it can actually be used, so having a better gun is meaningless if the fighter cannot reliably maintain its adversary in the weapon cone. The importance of having an advantage in maneuverability has been widely acknowledge in literature, and the existence of similar trends here provides mutual substantiation [135, 136].

Three sets of design attributes were considered for testing. The first corresponded to a highly-durable and highly-lethal system with baseline maneuverability characteristics. The second corresponded to a highly-maneuverable design which emphasized turn performance and low-speed capabilities with baseline durability and lethality. The third was a "utopia" design, possessing both the durability and lethality of the first set and the maneuverability of the second. The design variable settings for each case are given in Table 5.5.

Table 5.5: Design attribute values for testing

Case	λ_d	λ_t	r_g	θ_g	ω	P_S	u_{min}	u_{max}
1	1.4	0.6	1400	5	15	150	150	250
2	1.0	1.0	1000	4	22	150	125	250
3	1.4	0.6	1400	5	22	150	125	250
Units	$n.d.$	$n.d.$	m	deg	deg/s	m/s	m/s	m/s

Selection of Models

Two pairs of models were selected for generating trajectories for each case. The two pairs selected were those with the highest and lowest expected win probability for the designed fighter. The chosen design variable settings were not found in the LHS DOE, so surrogate models of win probability were created for each pair using ANNs trained with supervised learning. The DOE data was used for constructing the surrogate models, and an 80-20 split ratio was used for training and testing. Coefficients of determination for each ANN, as well as the predicted win probability for each case, are reported in Table 5.6. The model fits were reasonable, with coefficients of determination between 0.93 and 0.97. The range of predicted win probabilities for Cases 1 and 2 further highlighted how the analysis across a design space can be significantly influenced by the choice of behavior models employed.

Table 5.6: Coefficients of determination and predicted win probability for combinations of models and design attributes. Bold (italic) indicates highest (lowest) predicted win probability for that case.

Design Model	**Standard Model**	R^2	$P_{Win,1}$	$P_{Win,2}$	$P_{Win,3}$
23	8	0.9352	**0.8895**	*0.4148*	0.9862
23	2	0.9696	*0.1198*	**0.8794**	0.9748
21	8	0.9473	0.8831	0.6407	*0.9721*
21	2	0.9432	0.7608	0.5611	**0.9865**

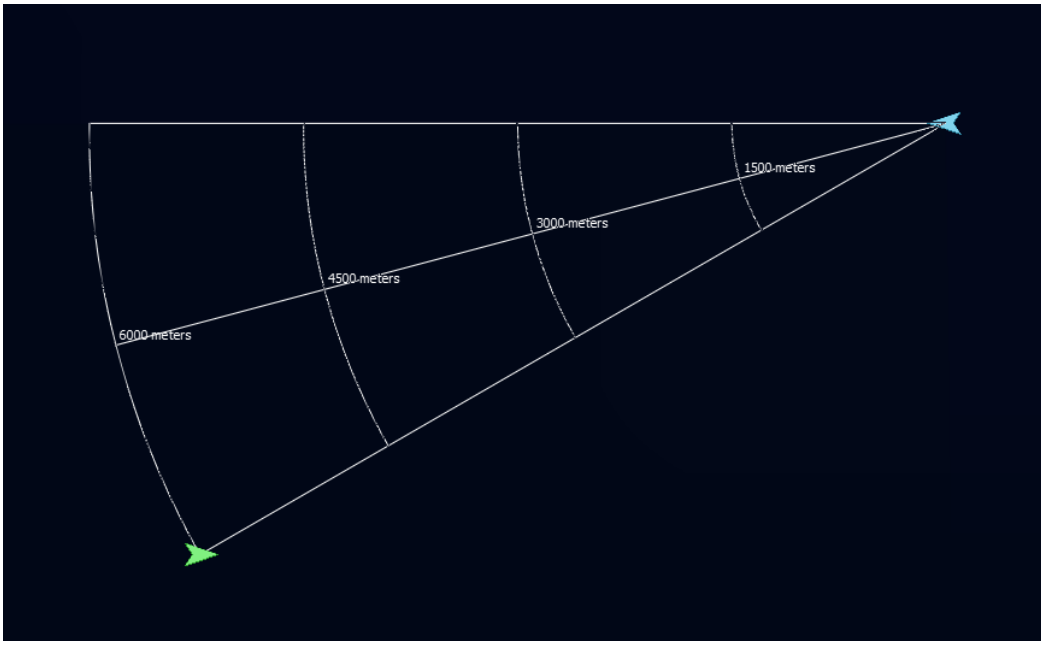


Figure 5.9: Initial geometry for testing. Here and in all future trajectories, the designed fighter is indicated by the blue wedge and the standard fighter by the green wedge.

Selection of a Test Scenario

A set of initial conditions had to be selected for generating trajectories for comparison. The scenario selected was based on observations on the depictions of fighter combat maneuvers described by Shaw, specifically the basic turning maneuvers for one-on-one engagements. Many of these are depicted as starting with the two fighters in a neutral position, facing one another at a range greater than their weapon engagement zones and with a small relative lateral offset. The initial separation between the fighters was set to 6 kilometers. The angle of the nose was set to 60 degrees, and the relative heading between them was set to 180 degrees. The initial geometry is shown in Figure 5.9, where the designed fighter is indicated by the blue wedge and the standard fighter by the green wedge.

Summary of Results

The designed fighters won four of the six test engagements, lost one, and tied one. The damage done by each fighter for each case is given in Table 5.7, where a value of 1 indicates

the fighter won the engagement. The designed fighters won in all cases where their model had the highest predicted probability of winning.

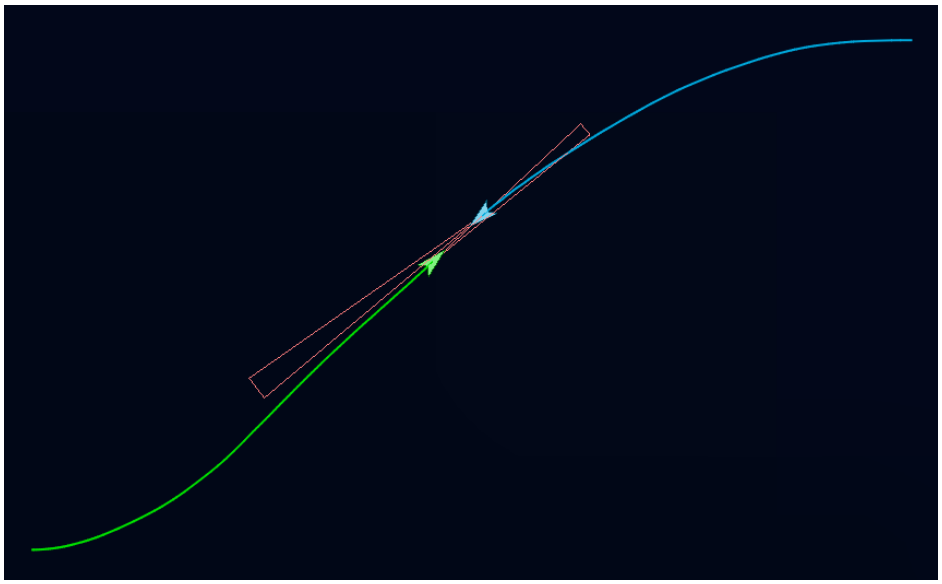
Table 5.7: Damage done by each fighter in test cases

Fighter	**Case 1**		**Case 2**		**Case 3**	
	Best	Worst	Best	Worst	Best	Worst
Designed	1.000	0.5931	1.000	0.0087	1.000	1.000
Standard	0.4340	0.7878	0.9662	1.0070	0.7203	0.7707

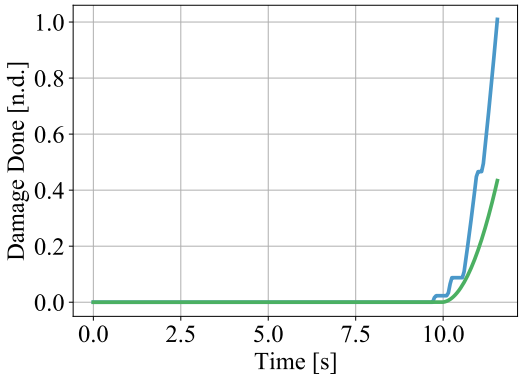
Case 1

The trajectory generated by Model 23 versus Model 8 for Case 1 design attributes is shown in Figure 5.10a, while the damage done to the opponent and agent speed are shown in Figures 5.11b and 5.11c, respectively. The two fighters clearly acted aggressively in this case, flying directly into one another and gaining speed to as to land high-scoring hits on their opponent as quickly as possible. The designed fighter had the advantage by virtue of its enhanced durability and lethality. However, this was clearly a high risk, high reward tactic. Minor deviations in flight path could easily move the target outside of the weapon cone and expose the fighter to excessive damage accrual and potential loss. This can be seen in the damage done plot, where there are small periods of time where the rate of damage done by the designed fighter goes to zero. It was saved solely the grace of its design attributes; lower damage output or higher susceptibility to incoming damage likely would have resulted in a loss if this strategy were used.

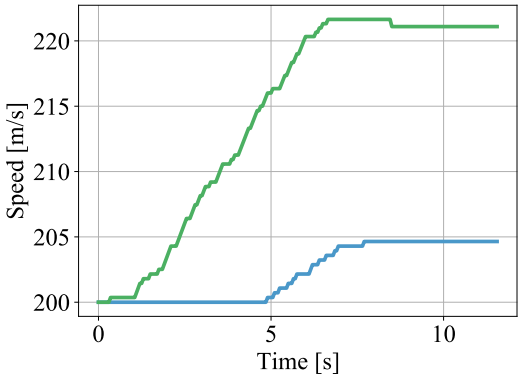
The trajectory, damage done versus time, and speed versus time for the model pairing which gave the designed fighter the lowest probability of winning in Case 1 are shown in Figure 5.11. The engagement began in a very similar manner to the other Case 1 simulation, with the both fighters turning towards one another. However, the fighters did not accelerate as dramatically in this case; the designed fighter gains some speed, but not enough to materially affect its turning capabilities. Because of this, and the fact that neither scored



(a) Trajectory



(b) Score vs time



(c) Speed vs time

Figure 5.10: Case 1, highest probability of winning

enough hits in the first pass to win, the two engage in a turning fight where neither can accord to disengage without putting itself at risk. The result is the two circling for the remainder of the engagement, making brief passes where neither is able to score without itself being hit, leading to a draw.

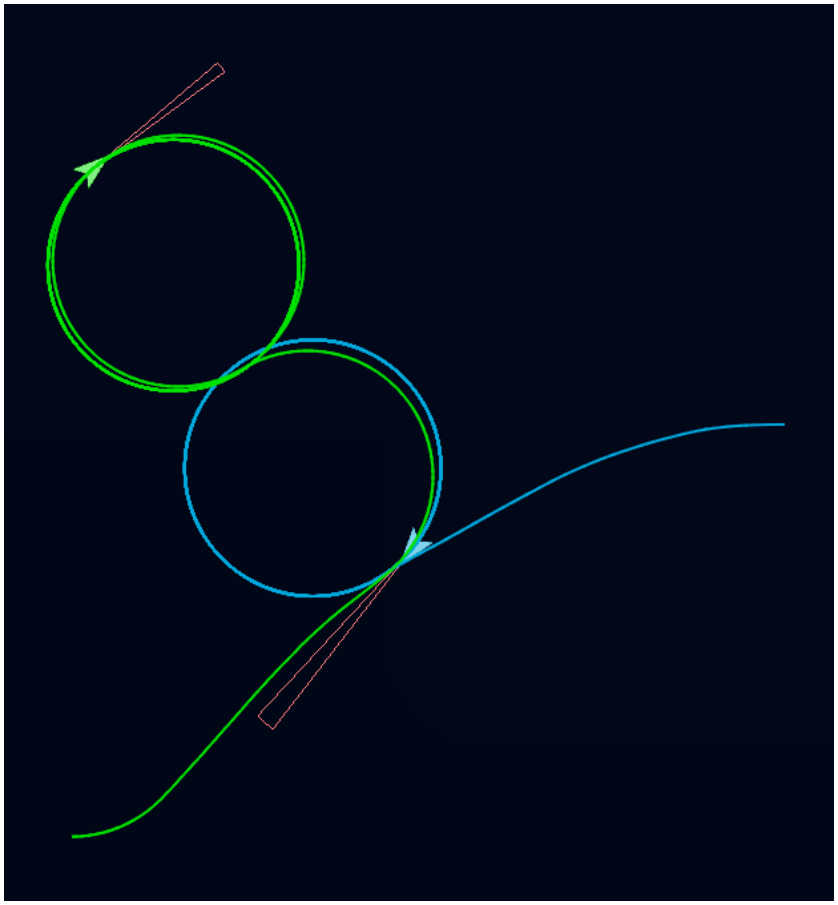
Case 2

The trajectory, damage, and speed data for the model pairing where the designed fighter had the highest predicted probability of winning the engagement are shown in Figure 5.12. The designed fighter came very close to losing this engagement after accruing a significant amount of damage on the first pass. However, it was able to survive just long enough to engage in a turning fight, and had bled a significant amount of speed going in. This allowed it to out-turn the standard fighter yet again, even though the latter had not accelerated going into the first pass. The maneuvers here were visually similar to those shown in Shaw's depiction of how fighters might disengage from the flat scissors, which is shown inset in the lower right corner of Figure 5.12a. There were minor differences but the general shape of the trajectories was remarkably similar.

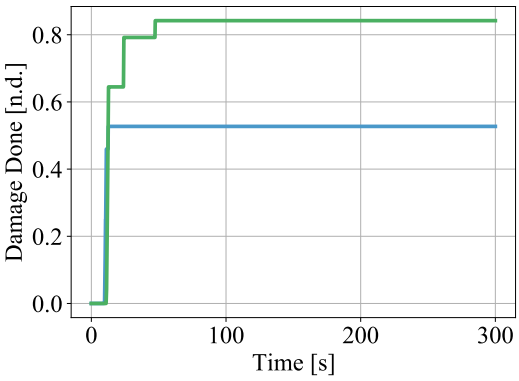
The results from simulating the pair of models for Case 2 where the designed fighter had the lowest predicted probability of winning are shown in Figure 5.13. The trajectory is very similar to that shown in Figure 5.10. However, the designed fighter in this case did *not* have the enhanced durability and lethality as previously, and so it loses the engagement without putting up much of a fight. It is never able to capitalize on its enhanced turning capabilities because it never gets into a turning fight, and losses the engagement going into the first pass.

Case 3

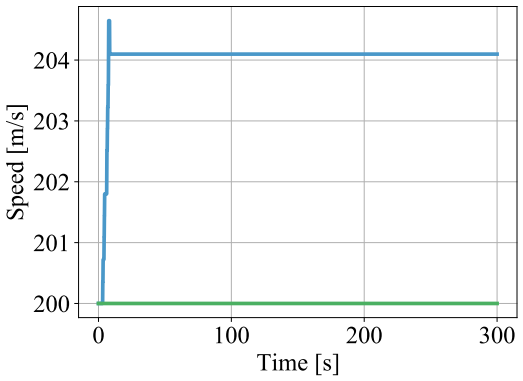
The engagements simulated under the conditions of Case 3 were, on average, more closely contested than any of the previous cases. The data on the pairs where the designed fighter



(a) Trajectory

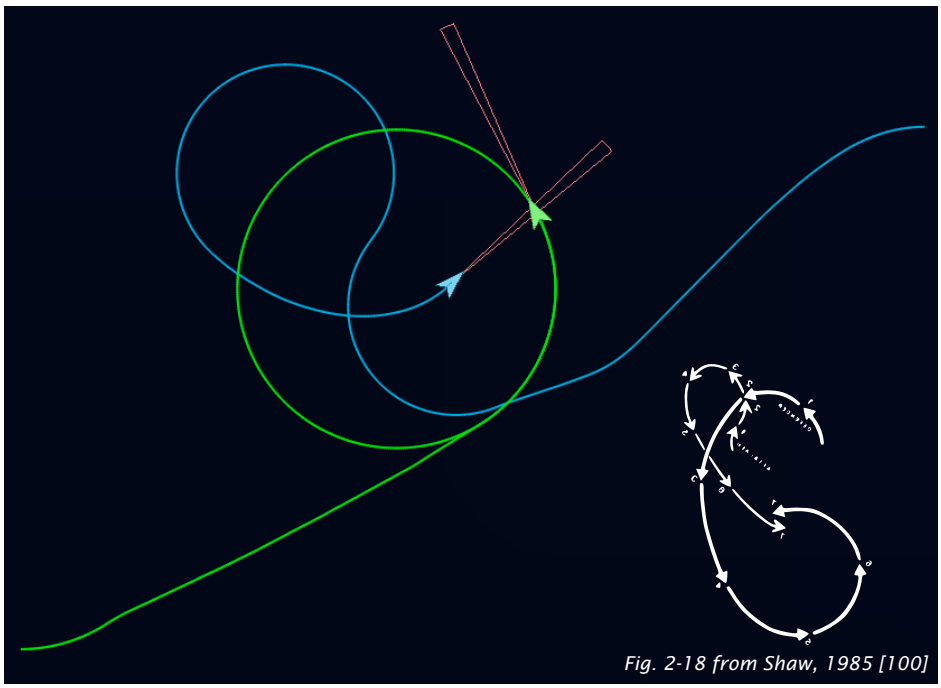


(b) Score vs time

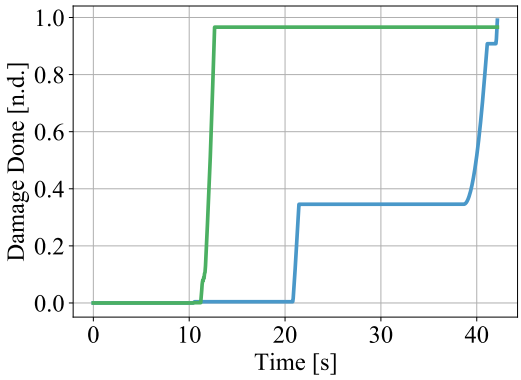


(c) Speed vs time

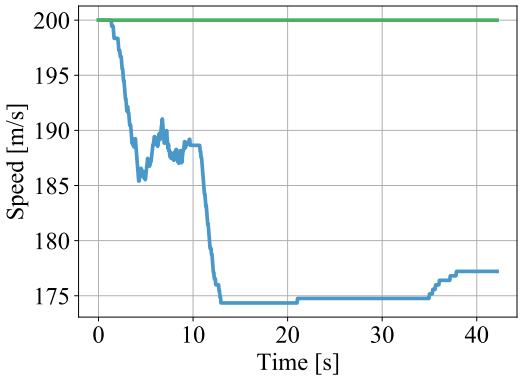
Figure 5.11: Case 1, lowest probability of winning



(a) Trajectory

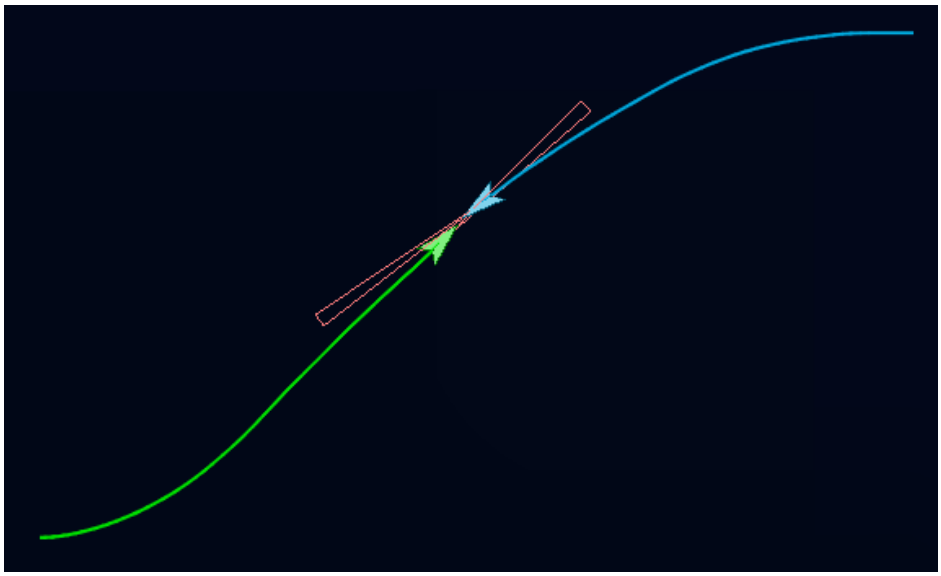


(b) Score vs time

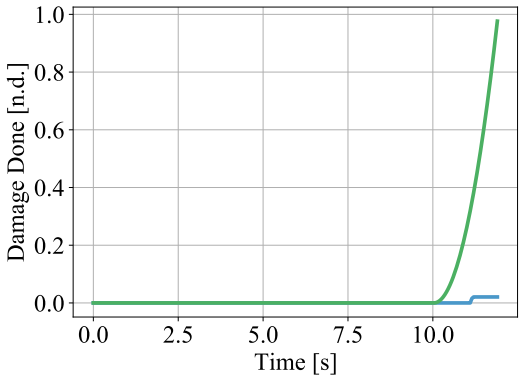


(c) Speed vs time

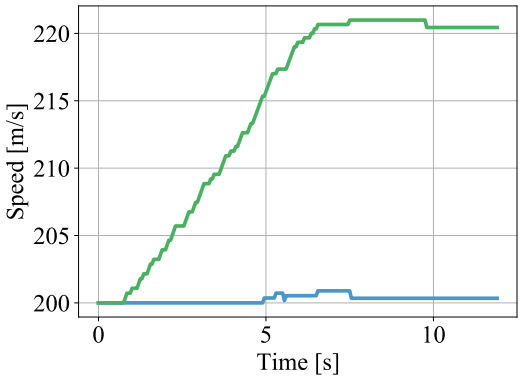
Figure 5.12: Case 2, highest probability of winning



(a) Trajectory



(b) Score vs time



(c) Speed vs time

Figure 5.13: Case 2, lowest probability of winning

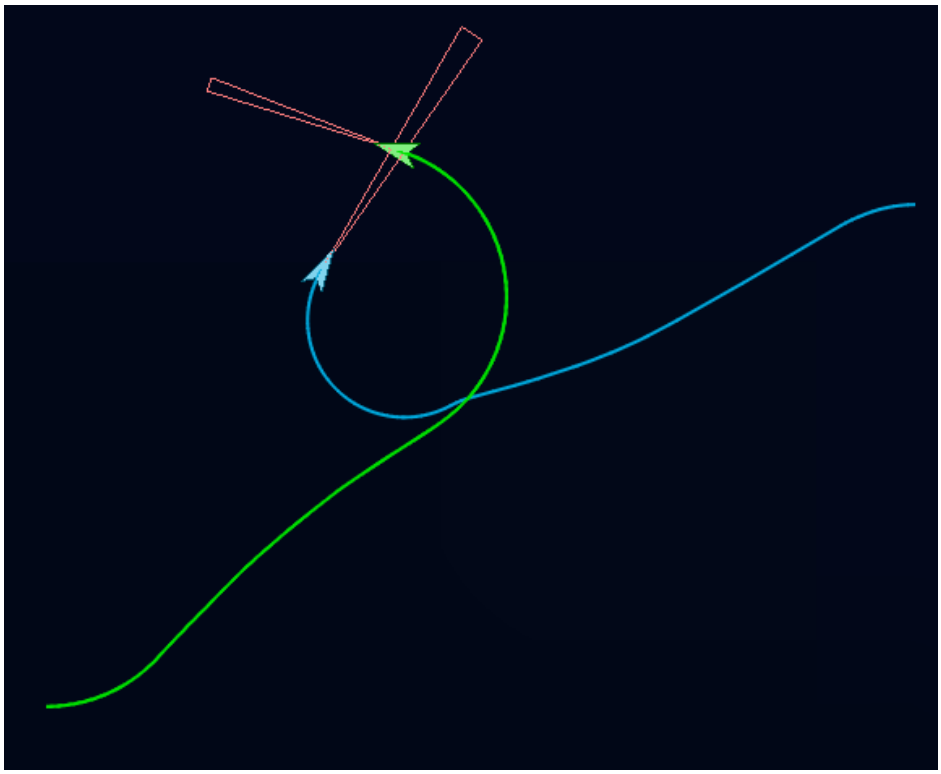
had the highest and lowest predicted probabilities of winning are shown in Figures 5.14 and 5.15, respectively. Both engagements became turning fights, similar to the previous cases. However, it appeared as though the designed fighter was leveraging *both* its enhanced durability and turn performance to achieve an earlier win while taking some risk in terms of damage accrual. It allowed its opponent to score hits in the first pass, relying on its low susceptibility parameter to survive through to the turning portion of the engagement. When turning, both models saw fit to maintain a lower speed than their opponent, enhancing their turning capabilities. The designed fighters then leveraged their higher turn rate to come around and land reliable, relatively close-range hits on their opponent and score a rapid win.

Recapitulation

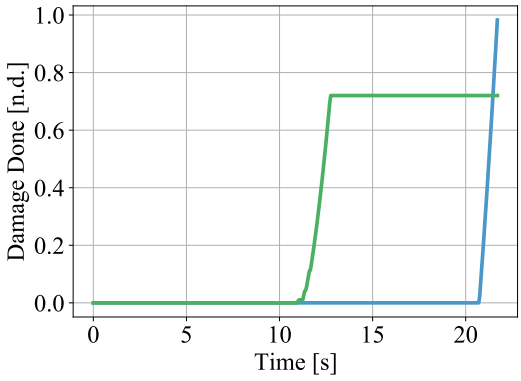
Visual inspection of the test trajectories showed the models had learned combinations of maneuvers similar to those seen in Shaw's *Fighter Combat: Tactics and Maneuvering*. The trajectories generated in turning engagements were visually similar to the flat scissors, with outcomes largely dictated by the speed and turn rate characteristics of each going into the maneuver. It was inferred that the models were engaging each other in distinct ways depending on the design variable settings, as evidenced by the corresponding changes to the trajectories. These inspections allowed for more-detailed insights into how the agents were behaving, which would have been unavailable at the level of statistical analysis and design space exploration used in the first part of the analysis effort.

5.9 Summary

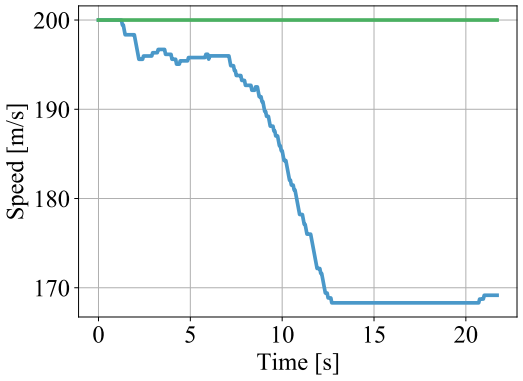
Application to the practical problem of designing a fighter aircraft for the purpose of winning a one-on-one, gun-only air combat engagement demonstrated the methodologies capacity to support design space exploration by enabling the exploration of employment concepts without the need to rely on human input. Models produced by the methodology



(a) Trajectory

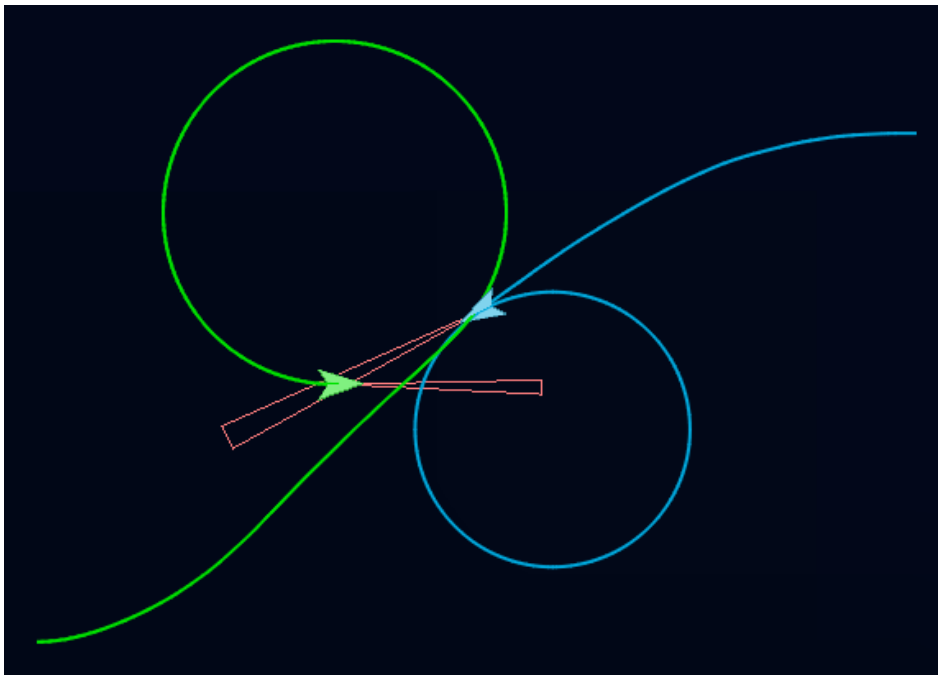


(b) Score vs time

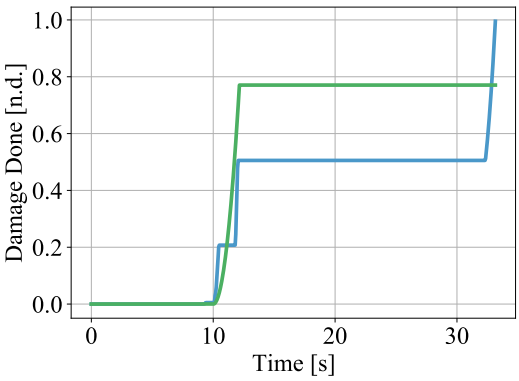


(c) Speed vs time

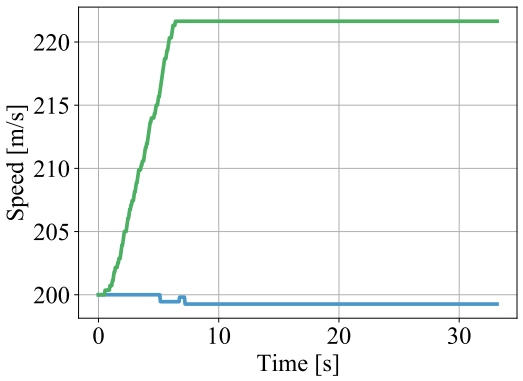
Figure 5.14: Case 3, highest probability of winning



(a) Trajectory



(b) Score vs time



(c) Speed vs time

Figure 5.15: Case 3, lowest probability of winning

were able to learn maneuvers which closely resembled documented fighter tactics honed over decades of human experiences: High throughput computing capabilities allowed those discoveries to happen in a matter of days.

There were two significant findings in the analysis of the design space using this methodology. Firstly, the well-known trends in the design space were largely confirmed. Winning dogfights has been widely recognized as a question of who can turn tighter, and that was seen here. The notion that having the capacity – and willingness – to rapidly decelerate in a turning fight was also observed in the visual inspection of test trajectories. Furthermore, it was shown that **the trends in the design space depended on the choice of models used to generate the evaluation data**. This was seen in previous experiments, and its repetition here further substantiates the need for explorations at the level of individual decision-making processes to support engagement-level analyses.

Secondly, while these findings were not surprising based on the existing knowledge base for this problem, it was important to recognize that the models had learned to map the large state space to decisive maneuvers entirely on their own. The interactions between relative position and relative speed had to be accounted for in the process of learning how to win an engagement against an opponent who was actively seeking the same end. At no point were the models instructed on *how* they were to behave; **the decision-making processes used in this quantitative evaluation were developed automatically**.

As with the previous experiments, there was certainly room for more training to be conducted for these models and the hyperparameters used during training could be tuned. However, the purpose of this experiments was primarily to demonstrate the applicability of the methodology, and **the results seen here support the overarching hypothesis**. The hypothesis was supported because the models were evidently able to explore the space of employment concepts within the amount of training allowed. Such explorations likely would not have been feasible or possible using input from human operators on the 1.6 million simulations used here.

CHAPTER 6
CONCLUSION

"I have already made this paper too long, for which I must crave pardon, not having now time to make it shorter."

—Benjamin Franklin

6.1 Review of Research Questions and Hypotheses

Five main research questions were identified in the course of this work. The first, **(RQ1) What type of modeling should be used to facilitate exploration and analysis of employment concepts?**, aimed to establish a basis upon which further research could be conducted. A review of possible modeling techniques showed computer modeling to be the most appropriate. This led naturally to another research question: **(RQ1.1): What type of computer modeling and simulation should be used?** This question was also answered through a review of available literature. In particular the works by Macal and North indicated the agent-based paradigm to be the most appropriate for enabling the kinds of explorations this research was primarily concerned with.

Further research in the uses of ABM arrived at the ODD and ODD+D protocols, which are community-standard methods for documenting the development and use of ABMs to support analysis. The 51 questions in the ODD+D protocol were created to facilitate model communication and establish confidence in the models used to inform decisions. In particular, several of the questions were intended to elicit information about the decision-making sub-models employed, what they were based on, and how they functioned. These protocol questions were distilled into focused research questions to be addressed by this work. First, the ODD+D protocol established the expectation that a sound theoretical or empirical basis be used in the creation and implementation of the agent decision-making models. Empirical

bases were excluded from consideration because including them could introduce unwanted bias, if they were even available in the first place since design space exploration could be used to evaluate technologies well beyond the bounds of past experiences. This led to the identification of control theory as a sound basis upon which further research could be based. An analogy was drawn between controllers and behaviors: Where the former regulates the activity of a system, behaviors regulate the activity of agents. However, optimal controllers can be very difficult to construct in the general case, so a generic controller construction process was distilled from available literature which could be used to inform the process for exploring and analyzing models of behavior.

The generic controller construction process was, fundamentally, one of experimentation, and experiments require both an apparatus upon which experiments can be conducted and an experimental process. The first question derived from this targeted research was: **(RQ2) How should state observations be mapped to actions?** This question sought to address a gap identified in the literature regarding the *how* of agent decision-making processes and what the experimental apparatus should. The next research question was: **(RQ3) How should state-action mappings be experimented with?** This was aimed at addressing a gap in the experimental process used to construct effective controllers. These two questions had to be addressed in tandem, since the apparatus and process could not be reasonably separated from one another.

There were two higher-level considerations which had to be brought to bear on the research. First, models of behavior rarely exist in isolation, and it is often the case that interactions between agents has a significant impact on the realized outcomes. Second, the larger design problem had to be considered, including how the changes in system design characteristics might influence the decision-making processes of that agent or others in the model. These considerations lead to the next two research questions: **(RQ4) How should explorations of employment concepts be conducted in models with multiple interacting agents?** and **(RQ5) How should explorations of employment concepts account for**

changes in design attributes?

A review of available literature allowed for the construction of a morphological matrix for addressing the stated research questions. Mathematical functions and decisions trees were identified from literature as possible techniques for mapping states to actions, and several numerical optimization techniques were identified to support the experimental processes. However, concerns were raised regarding the fitness of these techniques for the meeting the research objective. Both techniques for mapping states to actions had severe structural limitations, and the optimization techniques imposed artificial constraints on the exploration process which may be undesirable. Traditional optimization techniques also would not allow for adequate resolution of the credit assignment problem, which is a well-known problem within the subject of behavior modeling. Multidisciplinary design optimization was expected to be ill-suited to the problems under consideration by this research because the coupling between agent decision-making models could be extremely rigid, or it could expose agents to exploitation which would be difficult to recover from. Lastly, very little information could be found on techniques for enabling decision-making models to take design variables in account when mapping states to actions. The most relevant example found was Biltgen's work, where the design variable settings were included in the state space of a model to allow for better regression of expected performance as a function of the engagement parameters.

A broader literature search was conducted on the basis of the aforementioned observations with regard to existing techniques and methods. This led to the identification of artificial neural networks as a possible approach to addressing **RQ1**. ANNs would not have the same structural issues as mathematical functions or decision trees, and can be general function approximators, making them good candidates for this work. Further, reinforcement learning was identified as a candidate for exploring and experiment with the parameters of ANNs to improve performance and effectiveness. This made it a good candidate for addressing **RQ2**, although it could only be applied to ANNs and not to the other

forms of state-action mapping. The first and second hypotheses were stated in light these findings:

- **(H1) If artificial neural networks are used to map observable states to admissible actions then explorations of employment concepts will be made easier because the models will not be constrained by structural limitations**
- **(H2) If reinforcement learning is used to train artificial neural networks then explorations of employment concepts will be able to effectively address the temporal credit assignment problem because the effects of individual actions will be considered in the parameter optimization**

These hypotheses were tested on a representative problem – the pursuit-evasion scenario – and was shown to be more capable than existing, off-the-shelf models of behavior which are commonly used in literature. These empirical results substantiated **H1** and **H2** simultaneously.

Multi-agent reinforcement learning was next identified as a possible approach to addressing **RQ3**. MARL allows agents to learn *by interacting with one another*, possibly providing a direct solution to the challenge of exploring how those interactions might influence outcomes. The third hypothesis was stated in accordance with this: **(H3) If multi-agent reinforcement learning with multiple models per agent is used to train interacting agents in an engagement-level agent-based model then those agents will be able to learn robust and effective behaviors because the autocurriculum produced by interactions will enhance explorations**. This hypothesis was tested on the pursuit-evasion scenario, with separate ANNs controlling the pursuer and evader simultaneously. Two sets of tests were run: the first with a pursuer and evader model trained against each other for the full duration, and the second with a population of models for each agent which were randomly grouped for simulation and training at the start of each episode. Comparisons between the models showed the population-trained models performed roughly as well as the individually-trained ones, but it was also seen that the best individually-trained models

did not come from the same pairing. This indicated that the individual approach might not be able to produce results as reliably as the population approach. Further testing against the baseline algorithms showed both training methods produced robust models of behavior which were effective against opponent models they had never encountered during training, supporting **H3**.

Lastly, the augmentation of the observable state space with design variables was considered as a possible approach to allowing the behavior explorations to capitalize on or mitigate the potential benefits afforded by technologies. This was the basis of the fourth hypothesis: **(H4) If design attributes are treated as additional observable states then explorations of employment concepts will be able to consider the potential benefits afforded by variations in design attributes because they will be factored into the decision-making processes**. Two sets of models were trained, one without the state space augmentation and one with. Statistical comparison of performance showed that including the design variable settings in the state space allowed the models to learn better behaviors for their respective agents. A comparison of trajectories generated at distinct points in the design space showed the behaviors did not change significantly, but the models had learned to make use of their capabilities or mitigate the benefits of design changes, supporting **H4**.

The four hypotheses concerning individual components of the larger methodology were substantiated through experimentation and statistical analysis of empirical data. It was shown that these elements could provide the necessary capabilities to enable exploration of employment concepts in support of design space exploration. It should be noted that it was neither shown, nor demonstrated, nor claimed that these techniques were the only, nor the best options for doing so, only that they were fit for purpose and able to meet the needs of the research objective.

6.2 The StAR-Learn Methodology

The methodology of **St**ate Space **A**ugmentation for **R**einforcement **Learn**ing (**StAR-Learn**) was formulated for the purpose of enhancing design space exploration efforts by enabling explorations of employment concepts in concert technologies. Such explorations had historically relied on the input of subject matter experts or were simply left to the operators after systems had been designed and manufactured. Elements of the new methodology were shown to be capable of allowing those explorations to be moved upstream in the design process by leveraging advances in computational capabilities and reinforcement learning. A key contribution of this work was showing how design variables could be included in explorations of employment concepts to allow behaviors to be account for how those changes might influence the course of a simulation.

The ability to create highly effective models of behavior without the need for human input is expected to significantly enhance the design process by automating a labor-intensive portion of the analysis process. The methodology builds upon an established base and largely focuses on improving the process by which models are constructed to support quantitative evaluation. It was shown through statistical analysis of empirical data on a test problem that the proposed process can produce near-optimal models of behavior without the need to employ restrictive assumptions or complex analysis techniques.

The methodology was applied to the problem of designing a fighter aircraft for the purpose of engaging in one-on-one, gun-only air combat. Air combat tactics for these types of engagements have been the subject of intensive explorations by operators as aircraft have evolved over the past century. This provided an opportunity to test the methodology on a problem where some notion of effective models of behavior already existed in some form. Competing models learned effective engagement tactics starting from scratch, and were simultaneously exploring the design space around fighter capabilities. The results showed the models had learned to maneuver in ways which were strikingly similar to documented

fighter tactics available in open literature, namely the flat scissors and disengagement from it. Results from the design space exploration showed agreement with the general principles of air combat: The ability to achieve and maintain a positional advantage, or to deny the enemy such advantage, was critical to success. The results also showed how disadvantages in maneuverability could be mitigated by increasing the durability and lethality of the system, essentially demonstrating that one does not need to win the turning fight if they can win before it begins.

The results of applying the methodology to the air combat problem highlight an important idea at the heart of this research effort: **The choice of decision-making models used for design space exploration in engagement analyses can strongly influence the trends observed**. While it cannot be confidently stated that this methodology *solves* the problems of exploring employment concepts, the results shown here support the notion that the methodology is a feasible alternative for facilitating such explorations. The variety of models produced through use of **StAR-Learn** enable the adoption of multiple perspectives on the engagement problem from the standpoint of individual decision-making processes. Doing so increases the amount of information available to the analyst, even at early stages of the design process where system knowledge is scarce.

6.2.1 Potential Uses

StAR-Learn could be used in several places within the acquisition process. The first two experiments conducted in the course of this research demonstrated the potential for the methodology to aid engagement-level CBA by enabling analysis of existing, well-defined systems. The behavior models created by the inner methodology could aid in exposing potential gaps by exploring the various ways an engagement might unfold. The methodology could also be used on the DOTmLPF side of the fork after the creation of an ICD, specifically with respect to doctrine as an approach to closing the identified gaps.

The methodology could be used in support of higher-level analyses by providing more

information at the engagement level. Mission-level analyses could make use of the rich engagement data to inform broader explorations of existing or future capabilities. The third experiment showed how surrogate models of expected performance could be created using the data generated through an application of DSE using standard techniques. These surrogates could be used to expedite mission analyses by providing fast estimates of MOEs in support of e.g. Monte Carlo simulation. These surrogates could be implemented as parameters of the mission model or treated as random variables to further enrich the data generation process to support decision-making efforts.

6.3 Future Works

The work presented in this document in no way constitutes the end of the road for research into applications of reinforcement learning to problems of practical significance. The **StAR-Learn** methodology demonstrated a use case of RL, but several other implementations could be tested. Other RL algorithms could be tested, such as Soft Actor Critic, in an effort to determine the benefits and drawbacks to each. More sophisticated ANNs could also be tested. Recurrent, long short-term memory, and/or self-attention networks might provide additional benefits by allowing the effects of decisions and state observations to propagate forward in time. These types of investigations could help in gaining an understanding as to how well-suited different ANN architectures and training algorithms are to different classes of problems.

Further research into how the methodology could be utilized or extended to larger engagement-level analyses would be warranted. The engagement level covers one-on-one scenarios, which were investigated here, up to the softly-defined few-on-few scenarios. A relatively straightforward next step would be two-on-one and two-on-two engagements. Extensions to these higher order problems – which are still well-documented in the available literature – would help to advance trust in these novel techniques as more challenging problems are approached.

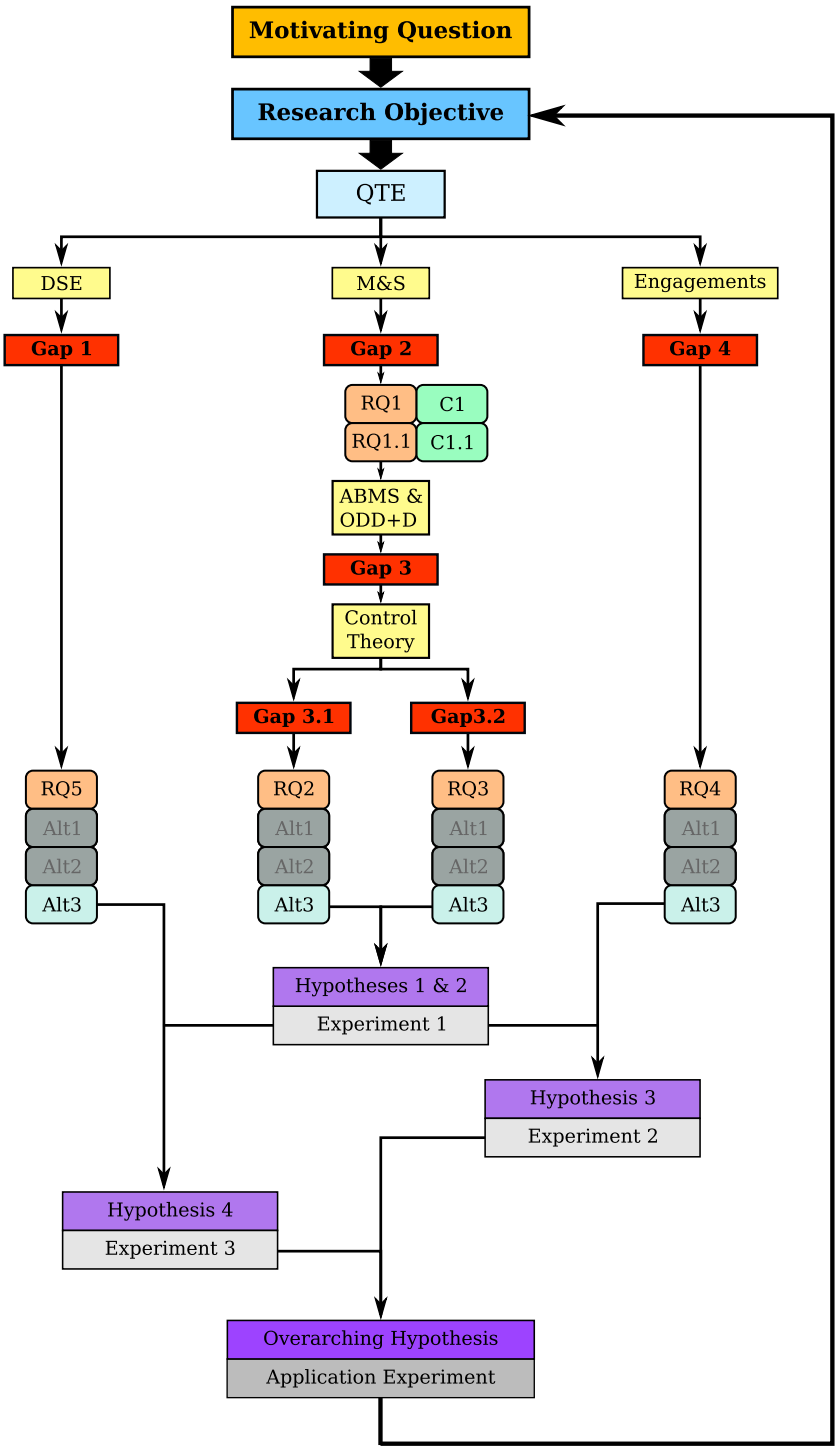
The experiments and application of the methodology focused on enhancing engagement-level analyses, which have a scope limited to scenarios with only a few entities. Development of an analogous mission-level methodology could prove useful, especially in light of the findings presented here. Alternatively, one could develop a distinct methodology to effectively leverage the results of engagement-level analyses produced by this one and propagate information up the analysis hierarchy. Mission- and campaign-level objectives are more abstract than those at the engagement level, necessitating some additional effort to translate those concepts into the RL paradigm, specifically with regard to the formulation of an appropriate reward mechanism.

A concept which was mentioned but not addressed in this work was that of explainable artificial intelligence. In general, the *reasoning* employed by an ANN in the course of its decision-making processes are largely a black box. Explainable AI seeks to address this by interrogating the models and convert their parameters into a form which can be more easily understood by humans. Such a capability would be invaluable in these types of analyses, where understandings *why* an agent acted a certain way could be leveraged in the design process and beyond. However, as a nascent area of research, it was not utilized here, leaving the matter open for future investigations.

Appendices

APPENDIX A

THESIS LOGIC DIAGRAM



History of Fighter Aircraft Design

F4F vs A6M | F-86 vs MiG-15 | F-4 | 5th Gen

Observation
Novel tactics have enhanced/mitigated technological disparities

Observation
Failure to account for interactions between tactics & technologies can adversely impact operational effectiveness

Observation
Tactical changes typically followed technological advances

Motivating Question
How can explorations of tactics be incorporated into the system design process?

System Design as a Decision Making Process

Establish Need
Define Problem
Establish Value Objectives
Generate Alternatives
Evaluate Alternatives
Make Decision

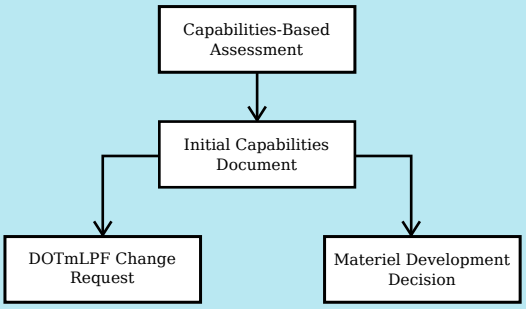
Observation
Need and problem derived from current or expected shortcomings

Observation
Capacity to meet value objectives will depend on how technology is used

Observation
Exploration of tactics would occur in generation and evaluation steps

Guiding Question
How are potential solutions to capability gaps generated and evaluated?

Closing Capability Gaps



Observation
Employment concepts are non-materiel solutions through doctrine

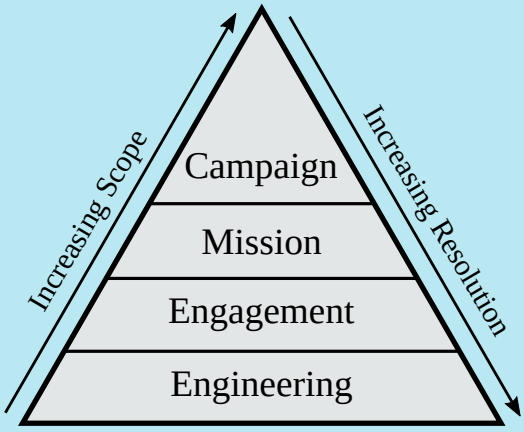
Observation
Employment concepts must be considered on both sides of ICD fork

Observation
New materiel solutions can be evolutionary or transformational

Observation
Evaluating employment concepts for new materiel likey more difficult

Context
This research focused on analyses of new materiel solutions to capability gaps

Analysis of Alternatives



Observation
Technologies are considered at the engineering level of analysis

Observation
Tactics (employment concepts) are considered at higher levels

Observation
Mission and campaign level are low resolution, broadly scoped

Observation
Analyses support one another up & down the hierarchy

Context
This research focused on engagement-level analyses because of their proximity to the technology design process in engineering analyses and use in supporting the other levels of analysis

Defining Doctrine

Observation
Tactics, employment concepts, and doctrine are similar concepts

Doctrine: The guiding **principles** regarding the employment and coordination of assets to achieve a common goal -- *JCIDS 2015*

Principle: A rule or belief governing one's **behavior** -- *Oxford English Dictionary*

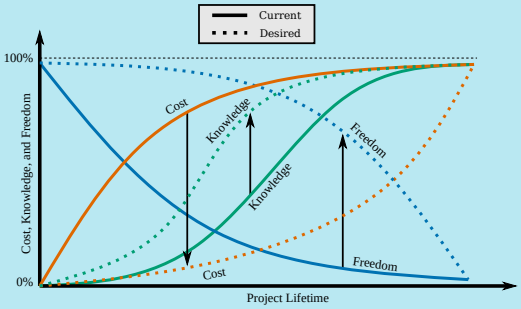
Behavior:
Anything an organism does that involves ... **response** to **stimulation**
-- *Wallace et al. 1991*
The way an organism **responds** to **stimulation**
-- *Raven & Johnson 1989*
A **response** to external or internal **stimuli**
-- *Starr & Taggart 1992*

Synthesized Representation of Behaviors

Stimulus → Principle → Response

Observation
Changes in behavior are realized through changes in principles

Challenges in Early Design Phases



Observation
Knowledge is limited in early design

Observation
Behaviors can increase design freedom

Observation
Failing to explore employment concepts might incur risk

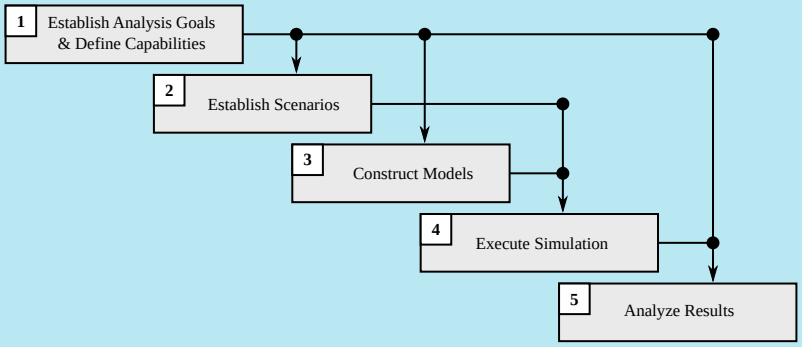
Research Objective

To enhance design space exploration for evolutionary and transformational solutions to capability gaps by enabling broader explorations of employment concepts to support engagement-level evaluations and analyses

Guiding Question

How can materiel solutions to capability gaps be generated and evaluated at the engagement level?

Quantitative Technology Evaluation



Observation
This process is a revision of generic decision-making process

Observation
Technologies defined in first step must be passed to behaviors

Observation
Generation of alternatives captured by model construction step

Observation
Employment concepts would be explored during model construction

There are three main components to the evaluation process:
1) Design Space Exploration -- Perturbing design variables to determine their effects on the value objectives
2) Modeling & Simulation -- The process used to generate quantitative data
3) Engagement Analysis -- Evaluating the system in the context of a relevant engagement scenario

Design Space Exploration

Observation
DSE is a process for "discovering and evaluating" designs

Observation
Begin with morphological matrix, consider different combinations of features

Observation
Morphologies can be categorically different

Observation
Detailed evaluation can be conducted once a morphology has been selected

Alternative / Characteristic	1	2	3	4
Vehicle	Wing & Tail	Wing & Canard	Wing, Tail, & Canard	Wing
Minimum Combat Radius (km)	500	1000	1500	
Payload (kg)	6,000	8,000	10,000	
Maximum Mach	1.2	1.6	2.0	
Ceiling (km)	10	15	20	
Observability	Low	Standard		
Armament	Guns	Missiles	Guns & Missiles	
Weapon Bays	Internal	External	Internal & External	

Scoping
The DSE in this work was confined to the detailed analysis of a single morphology

Observation
DSE on a single morphology involves perturbation of design attributes and observation of effects on metrics

Observation
The number of points generated for DSE can be large

Observation
It would not be reasonable to assume the effects of design variables on decision-making processes are known

Gap 1
The potential effects of design attributes must be considered when exploring employment concepts

Observation
This gap can only be addressed after subsequent, lower-level elements of the process have been established

Modeling & Simulation

Observation
M&S is used to develop concepts, aid in answering questions

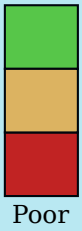
Observation
M&S can be used to facilitate analysis and evaluation

Gap 2
An appropriate modeling paradigm for exploring employment concepts is needed

Observation
There are many different types of M&S

Research Question 1
What type of modeling should be used to facilitate exploration and analysis of employment concepts?

Criterion \ Alternative	Physical Modeling	Conceptual Modeling	Regression	Computer Modeling
Appropriate for Early Design Evaluation				
Quantitative				
Facilitate Explorations of Behaviors				
Mitigate Bias				
Reasonable Cost				



Conjecture 1
Computer M&S is the most appropriate paradigm for this work

Research Question 1.1
What type of computer M&S should be used?

When and Why ABMS -- Macal & North, 2005
-- There is a natural representation as agents
-- Agents adapt and change their behaviors
-- Agents learn and engage in dynamic strategic behaviors
-- Agents have dynamic relationships with other agents
-- The past is no predictor of the future

Conjecture 1.1
Agent-based modeling & simulation should be used

Behaviors in Agent-Based Modeling

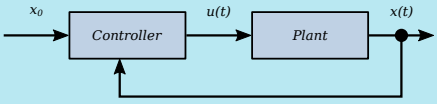
Observation
Behaviors must be implemented using computer methods

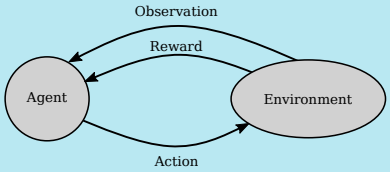
Observation
ODD+D protocol proposes criteria for modeling human decision-making processes

Observation
Adequate justification for the choice(s) of behavior models is needed to establish confidence in results, data

Gap 3
A theoretical foundation for exploring and analyzing employment concepts is needed

Guiding Question
What is an appropriate theoretical basis for behavior modeling?





Observation
Agents and their behaviors can be viewed as controllers

Observation
Literature on optimal control theory provides a template for constructing optimal controllers

#	Step
1	Describe the system
2	Identify observable states
3	Identify admissible controls
4	State performance index
5	Experiment with controls
6	Select a best controller

Observation
Constructing models is a process of experimentation

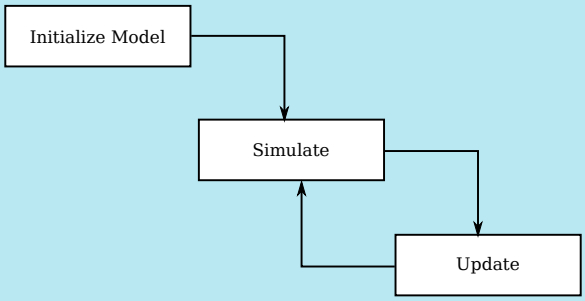
Experimenting with Behaviors

Guiding Question
How should behavior models be experimented with?

Observation
Skinner established a generic process for learning new behaviors based on experience

Observation
Learning by experience involves acting, observing effects, and updating behaviors to increase desirability of outcomes

Observation
These types of experiments require two components:
An apparatus to experiment on and an experimental process



Observation
Attributing credit/blame over sequences of actions can be difficult (temporal credit assignment)

Observation
Number of possible decision "paths" in a sequence grows exponentially with time (curse of dimensionality)

Gap 3.1
A technique to allow agents to map observed states to admissible actions is needed is needed

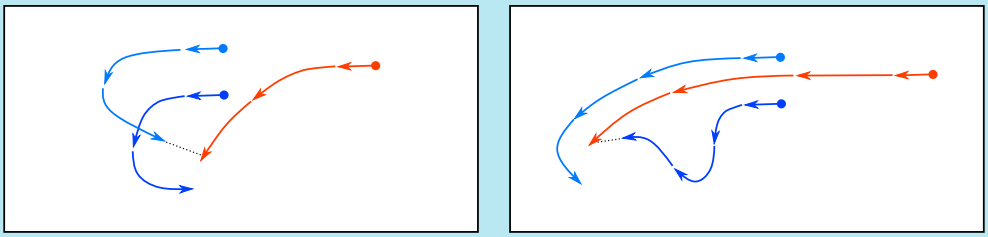
Gap 3.2
A process for exploring and evaluating different state-action mappings is needed

Engagement-Level Analyses

Guiding Question
How can behaviors be explored for interacting agents?

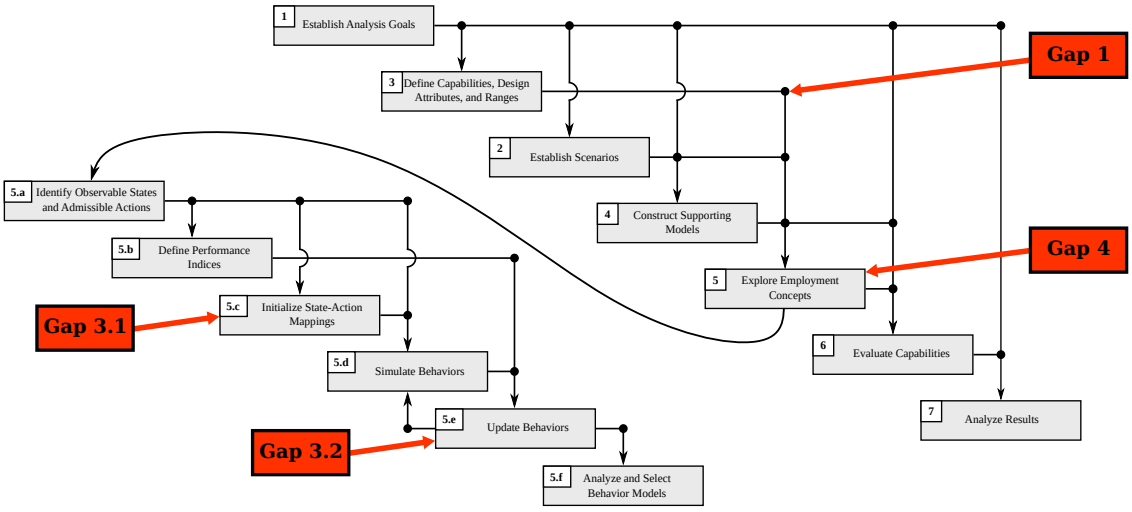
Observation
Interactions exacerbate the curse of dimensionality and credit assignment problems

Observation
Two versions of half-split maneuver show how interactions can influence evolution of simulation and outcomes



Gap 4
A technique for facilitating exploration and evaluation of employment concepts for multiple interacting agents is needed

Summary of Gaps
Basic quantitative technology evaluation process expanded based on findings from literature; gaps remain to be filled by further research



Gap 3.1: Experimental Apparatus

Observation
The stimulus-principle-response representation of behaviors is analogous to the concept of a function

Observation
The purpose of the function is to make decisions

Observation
Might not know form of decision-making process at start

Alternative Mathematical Functions
-- Used in optimal control
-- Transparent evaluation
-- Must select finite terms
-- Continuous actions only

Alternative Decision Trees
-- Used in ABMS
-- Traceable evaluation
-- Must select finite nodes
-- Discrete actions only

Gap 3.2: Experimental Process

Observation
Behavior models have two dimensions which can be experimented with: structural and parametric

Observation
Structural experiments might be more difficult, less justifiable

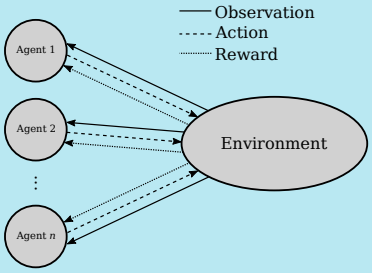
Observation
Parametric experimentation is akin to numerical optimization

Alternative First Order Methods
-- Ensures improvement
-- Only local guarantees
-- Gradients can be costly
-- Not applicable to DTs

Alternative Zeroth Order Methods
-- Applicable to all models
-- Good exploration
-- High cost
-- Potential limits on values

Gap 4: Engagements

Observation Distinct agents might have distinct objectives
Observation Agents can compete or cooperate in engagements
Observation Desirability of certain states, actions depends on behavior of other agents in the scenario model



Alternative Multi-Objective	**Alternative** Multi-Disciplinary
-- Explore simultaneously -- Find family of solutions -- Large problem space -- High computational cost	-- Solve individually -- Find singular solution -- Possible exploitation -- Difficult for stiff models

Gap 5: Design Space Exploration

Observation Limited literature on behavior exploration in DSE
Observation Biltgen showed decision-making could be influenced by including design variables in state space
Observation Some behaviors might be independent of design variables

Alternative Robust	**Alternative** Partitioned	**Alternative** Augmented
Explore behaviors independent of design variable settings	Explore behaviors independent of design variable settings across small regions of space	Include design variable settings in observed state

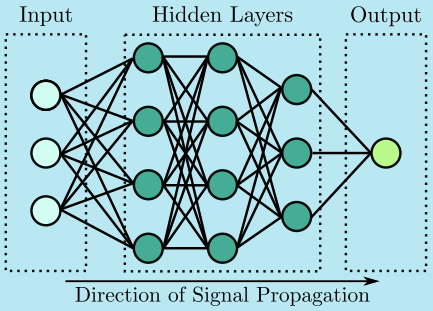
Summary of Existing Capabilities

Characteristic \ Alternative	1	2	3
State-Action Map	Mathematical Function	Decision Tree	
Experimentation & Exploration	First Order Optimization	Zeroth Order Optimization	
Engagement-Level Analyses	Multi-Objective	Multi-Disciplinary	
DSE	Robust	Partitioned	Augmented

Observation
Several challenges in experimental apparatus & process

Observation
Engagement and DSE alternatives have not been tested

An Alternative Approach



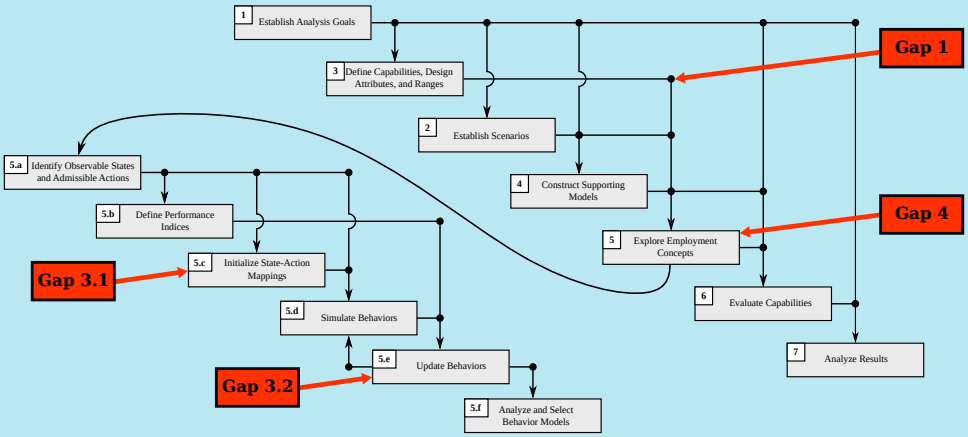
Observation
Artificial neural networks can be used to map states to actions

Observation
ANNs are general function approximators, mitigating structural concerns

Observation
Reinforcement learning for training ANNs is similar to operant conditioning

Observation
Multiagent RL has been used to train agents in competitive & cooperative games

Hypothesis Composition



Characteristic \ Alternative	1	2	3
Gap 3.1 State-Action Map	Mathematical Function	Decision Tree	Artificial Neural Network
Gap 3.2 Experimentation & Exploration	First Order Optimization	Zeroth Order Optimization	Reinforcement Learning
Gap 4 Engagement-Level Analyses	Multi-Objective	MDO	MARL
Gap 1 DSE	Partitioned	Robust	Augmented State Space

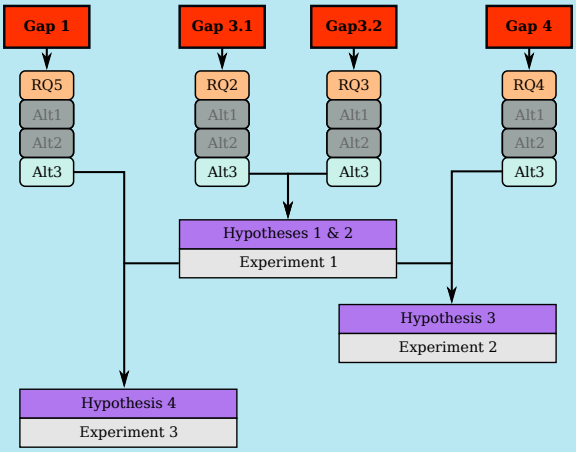
Hypothesis 1
If artificial neural networks are used to map observable states toadmissible actions then broader explorations of employment concepts will be possible because the models will not be constrained by structural limitations

Hypothesis 2
If reinforcement learning is used to train artificial neural networks then effective exploration of employment concepts will be possible because individual actions will be considered, mitigating the credit assignment problems

Hypothesis 3
If multi-agent reinforcement learning with multiple models per agent is used to train interacting agents in an engagement scenario then those models will be able to learn effectivebehaviors because the more diverse autocurriculum will enable broader exploration

Hypothesis 4
If design attributes are treated as observable states then thetrained behavior models will be better able to mitigate orcapitalize on different settings because the design attributes willbe factored into the decision-making processes

Hypothesis Testing



Observation
Techniques have not been proven or tested for relevant problems

Observation
Bottom-up demonstration of individual elements can build confidence

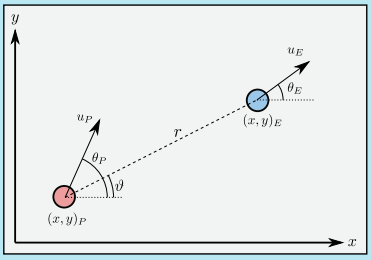
Observation
Need an appropriate test problem

Problem Criteria:
-- Multiple interacting agents with related objectives
-- Performance affected by changes in design attributes

Experiment Criteria:
-- Clearly defined measures of performance and effectiveness
-- Examples of behavior exist in literature
-- Low computational cost

Observation
The pursuit-evasion scenario meets all stated criteria

Pursuit-Evasion
-- Two entities, a pursuer P and evader E
-- P trying to capture E
-- E trying to avoid, delay capture
-- Time to capture depends on speed of each
-- Whether E is captured depends on behaviors



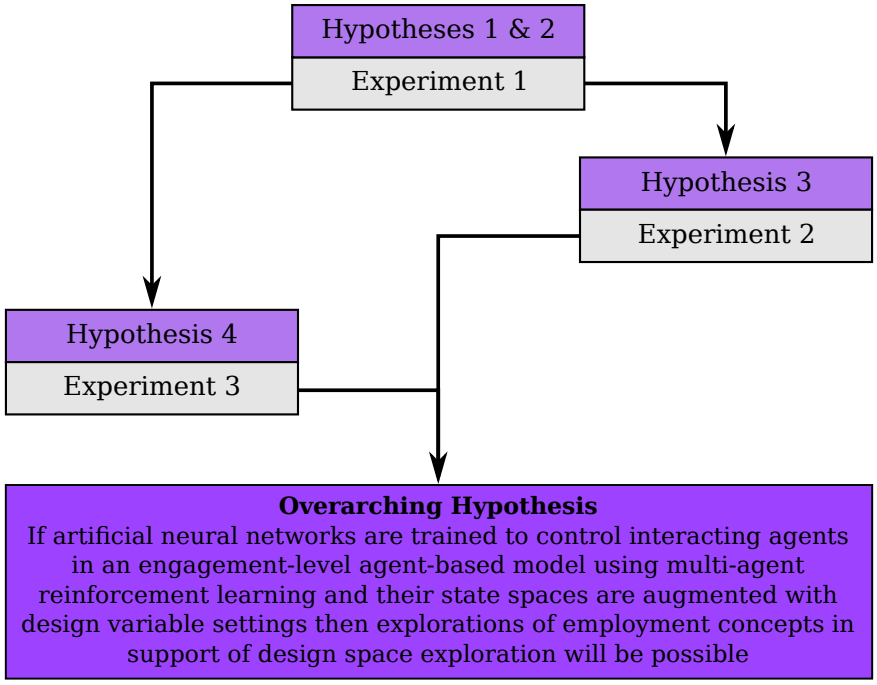
Results of Experiments

Observation
ANNs trained using RL performed better than baselines

Observation
MARL allowed agents to learn simultaneously

Observation
ANNs trained with MARL achieved high levels of performance against baselines without being trained against them

Observation
Augmenting the state space improved performance against baselines across the design space



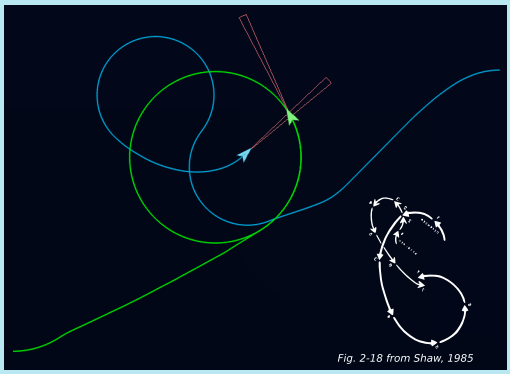
Application Experiment

Observation
One-on-one, gun-only air combat engagements meet the problem criteria established at the outset of experimentation

Observation
One-on-one, gun-only air combat engagements are reasonably well-understood and example tactics are available in literature

Observation
Desirable design characteristics for these types of engagements have been established through experience

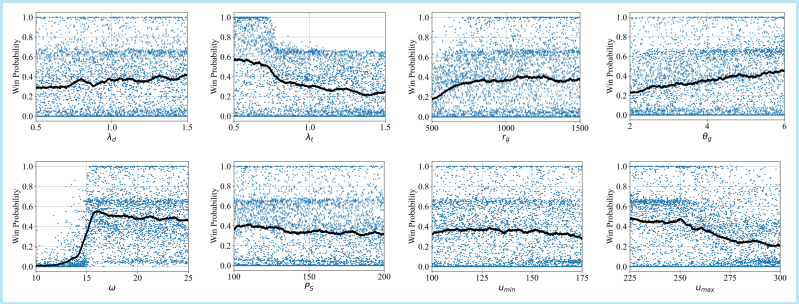
Observation
Desirable design characteristics for these types of engagements have been established through experience



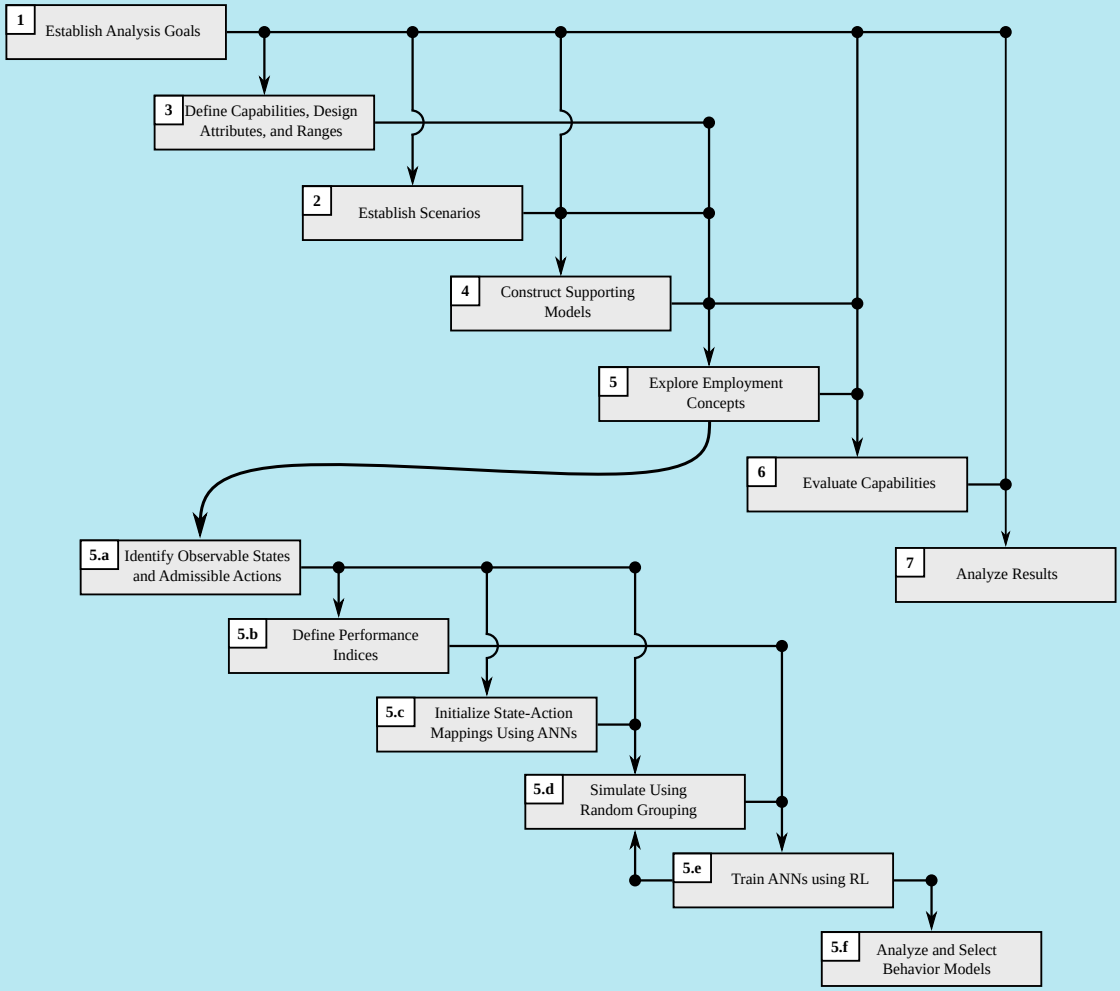
Observation
ANNs learned effective maneuvers to acheive their goals

Observation
ANNs exhibitied behaviors similar to established maneuvers

Observation
Choice of trained ANN influenced trends in design space and range of design points for which value objective was met



Conclusion: State Space Augmentation for Reinforcement Learning (StAR-Learn)



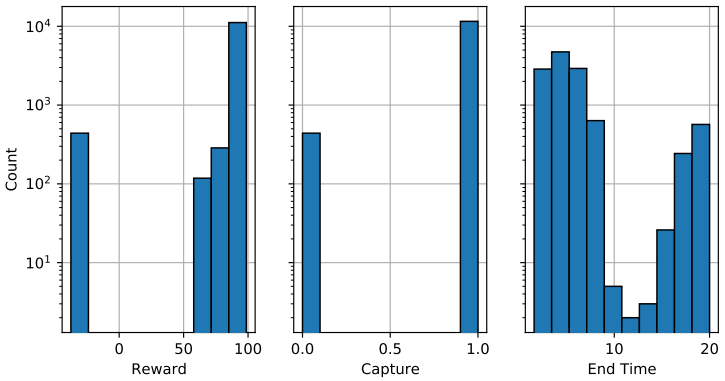
Synthesis of the identified techniques enabled simultaneously exploration of employment concepts and design spaces

The proposed methodology was shown to be capable of satisfying the research objective

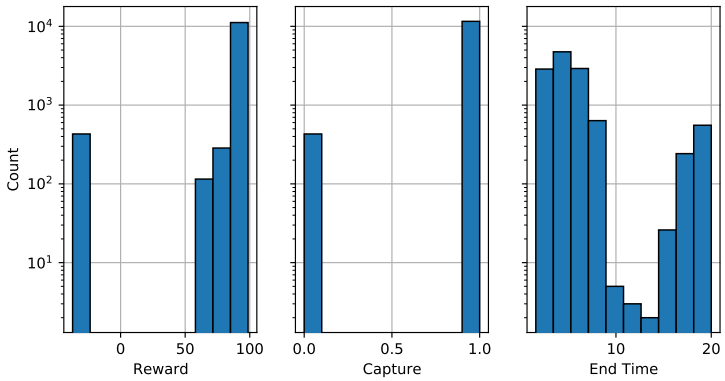
APPENDIX B

SUPPLEMENTARY DATA VISUALIZATIONS

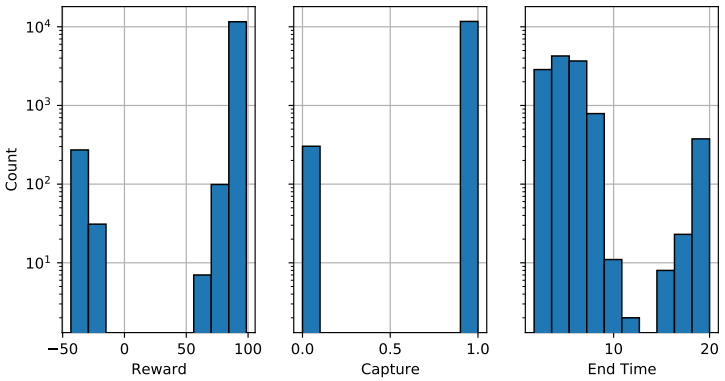
This appendix presents supplementary figures supporting the observations and arguments derived from the experiments in Chapters 4 and 5.



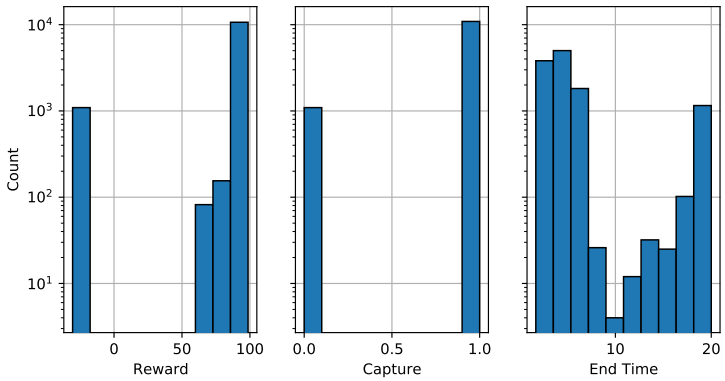
(a) Versus PE



(b) Versus BE

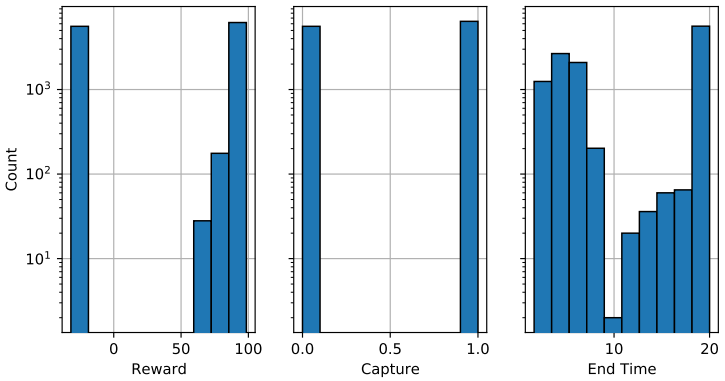


(c) Versus PE

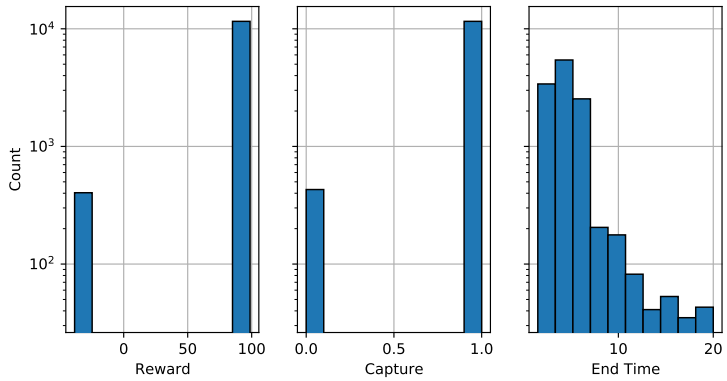


(d) Versus BE

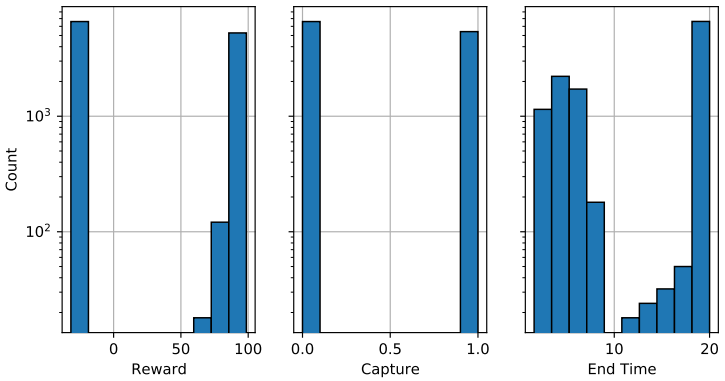
Figure B.1: Distributions of performance metrics for Population pursuers



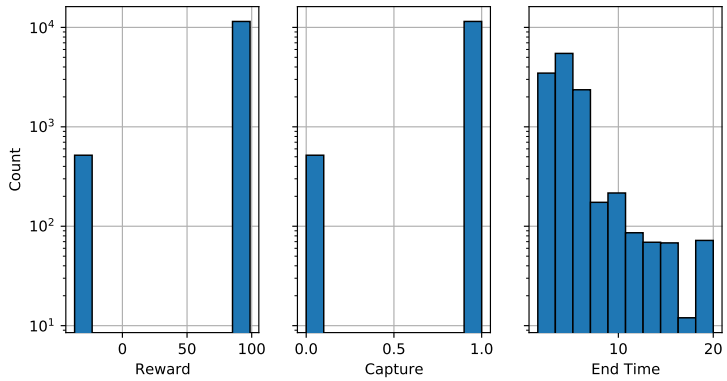
(a) Ind vs PP



(b) Ind vs PN

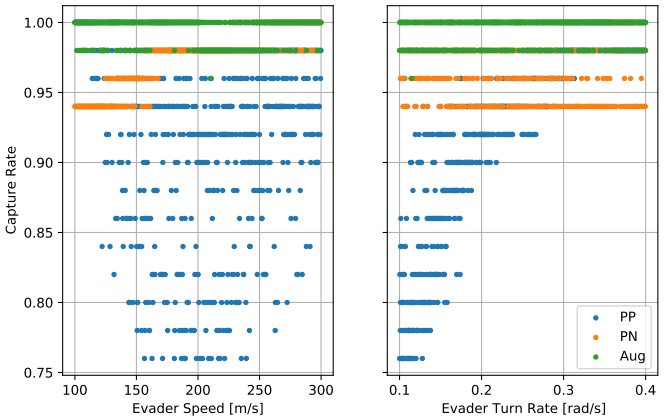


(c) Pop vs PP

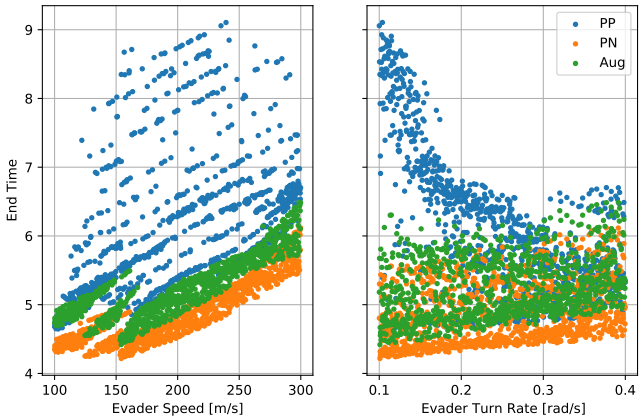


(d) Pop vs PN

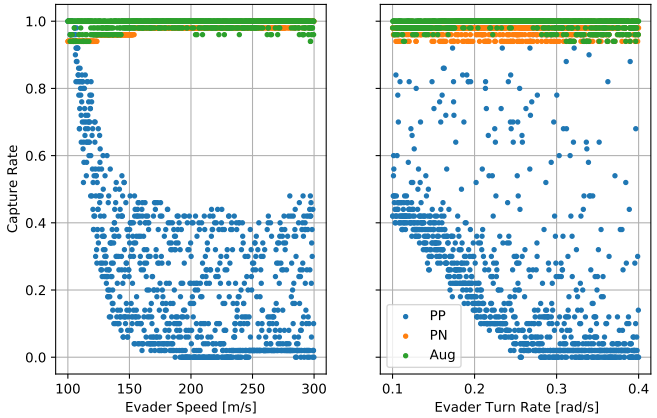
Figure B.2: Distributions of performance metrics for evaders



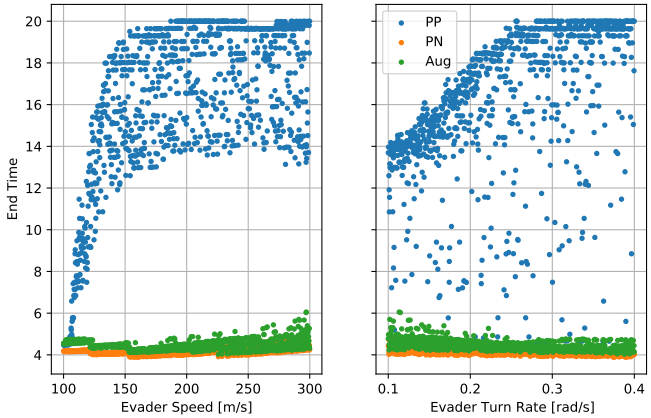
(a) Capture



(b) End Time

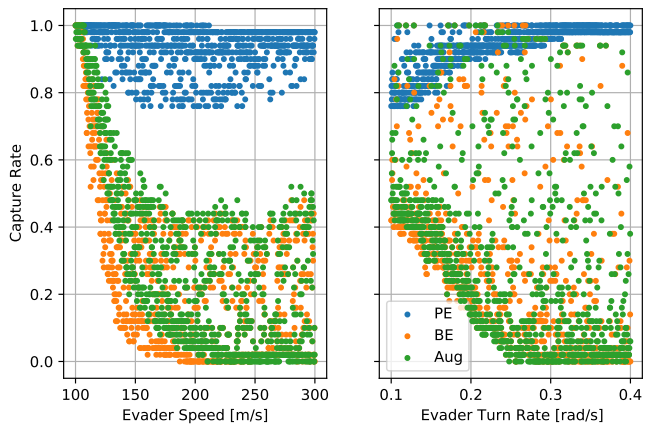


(c) Capture

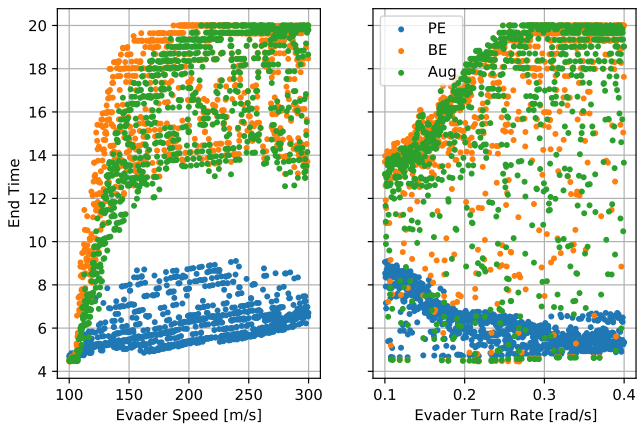


(d) End Time

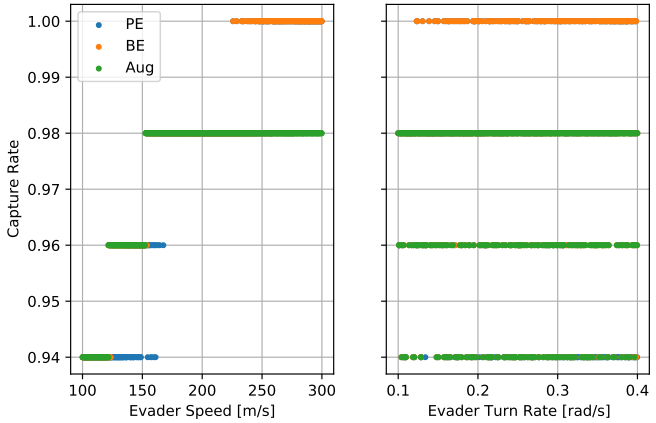
Figure B.3: Comparison of pursuer metrics versus baseline evader guidance algorithms



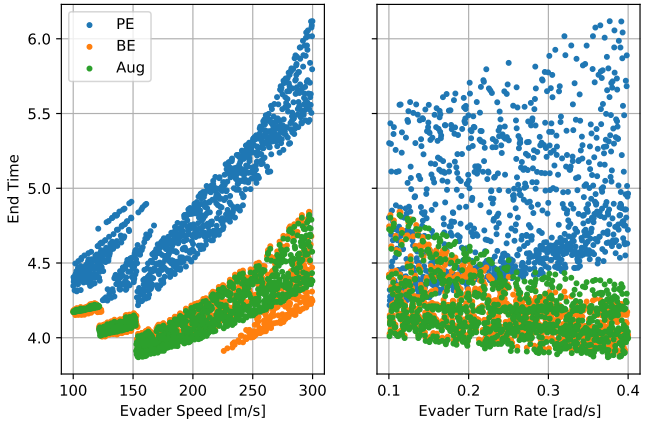
(a) Capture vs PP



(b) End Time vs PP

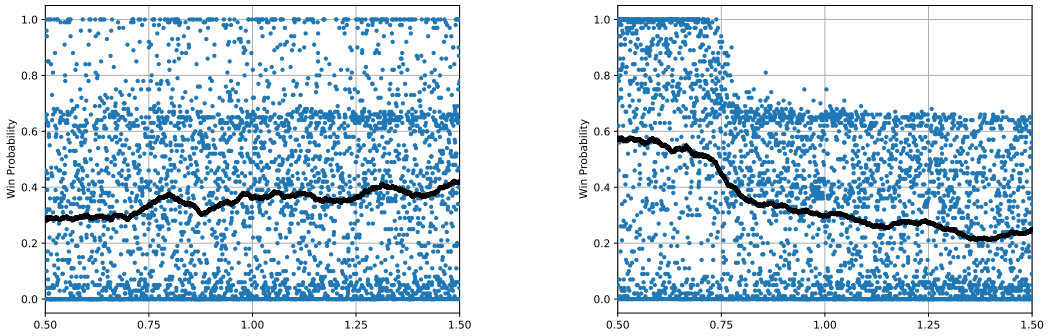


(c) Capture vs PN



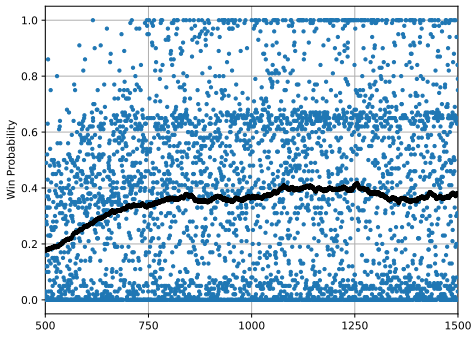
(d) End Time vs PN

Figure B.4: Comparison of evader metrics versus baseline pursuer guidance algorithms

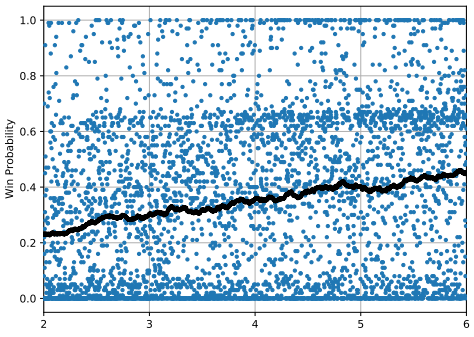


(a) Versus Lethality

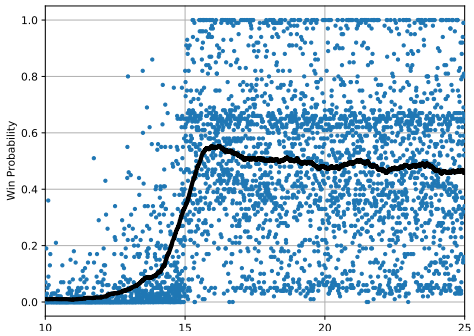
(b) Versus Susceptibility



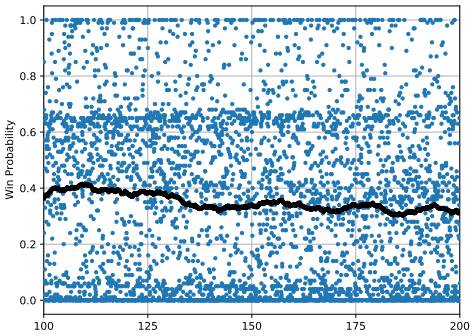
(c) Versus Gun Range



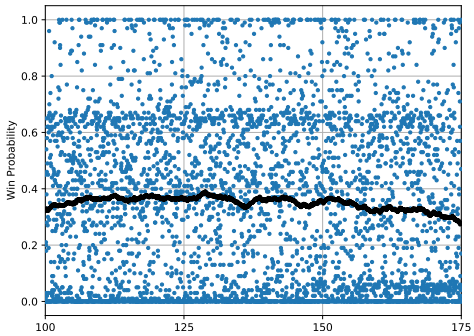
(d) Versus Gun Angle



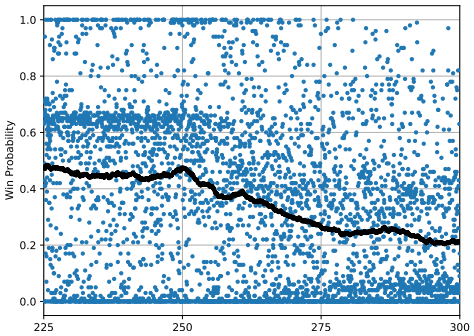
(e) Versus Turn Rate



(f) Versus Excess Power

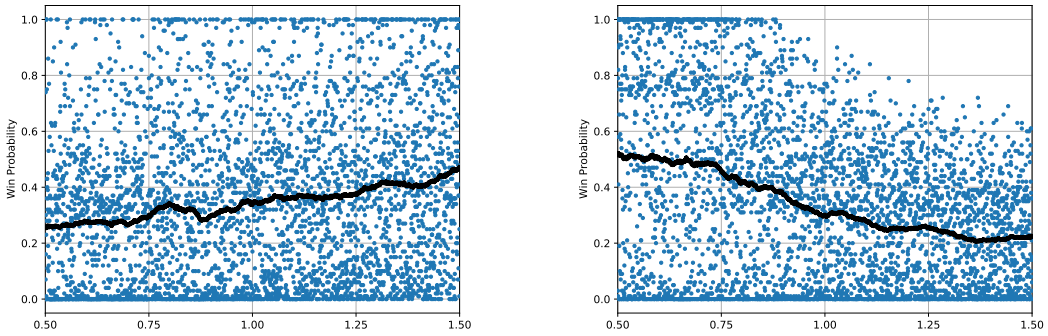


(g) Versus Minimum Speed



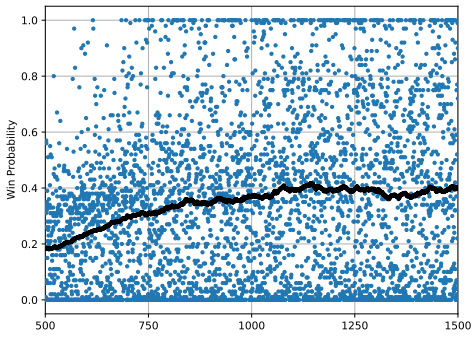
(h) Versus Maximum Speed

Figure B.5: Win probability for designed fighter versus design variable settings. Designed fighter was controlled by Model 21, standard fighter by Model 8

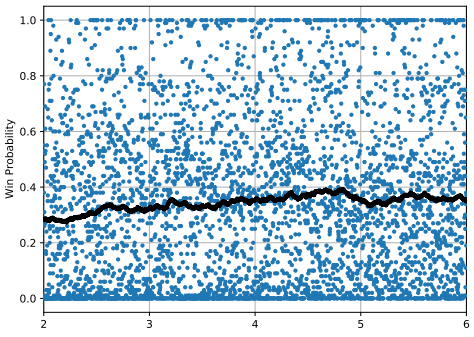


(a) Versus Lethality

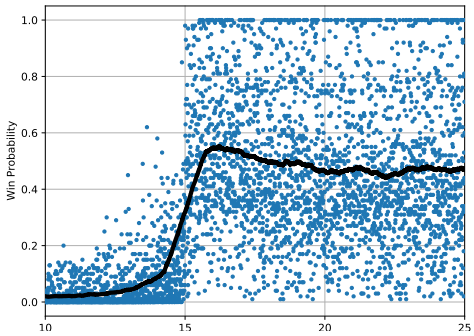
(b) Versus Susceptibility



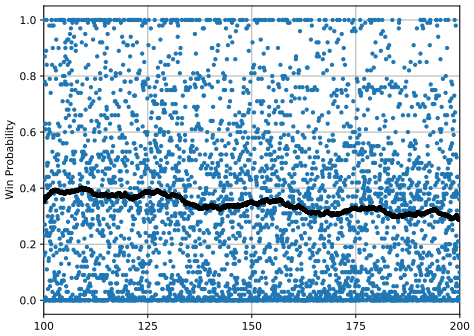
(c) Versus Gun Range



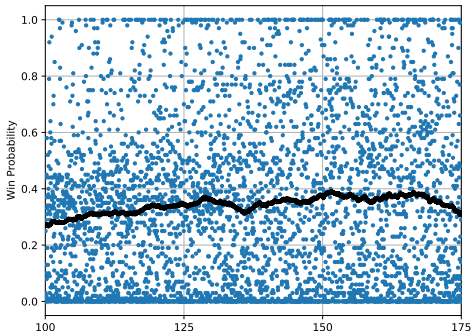
(d) Versus Gun Angle



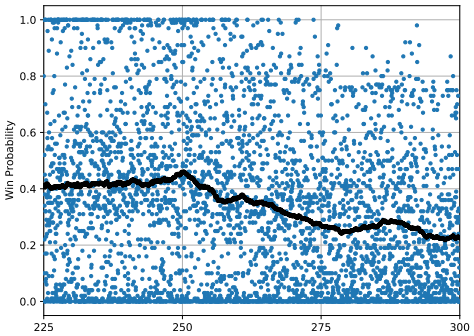
(e) Versus Turn Rate



(f) Versus Excess Power



(g) Versus Minimum Speed



(h) Versus Maximum Speed

Figure B.6: Win probability for designed fighter versus design variable settings. Designed fighter was controlled by Model 21, standard fighter by Model 2

REFERENCES

[1] S. Abar, G. K. Theodoropoulos, P. Lemarinier, and G. M. O'Hare, "Agent based modelling and simulation tools: A review of the state-of-art software," *Computer Science Review*, vol. 24, pp. 13 –33, 2017.

[2] D. S. Alberts, *Code of Best Practice: Experimentation*. Jul. 2002.

[3] D. S. Alberts and R. E. Hayes, *Code of Best Practice: Campaigns of Experimentation, Pathways to Innovation and Transformation*. Jan. 2005, p. 257.

[4] "Algebraic.". (2021). Merriam-Webster.com, (visited on 04/23/2021).

[5] R. J. Allan, "Survey of agent based modelling and simulation tools," Daresbury Laboratory, Tech. Rep., 2010.

[6] C. R. Anderegg, *Sierra Hotel: Flying Air Force Fighters in the Decade After Vietnam*. DIANE Publishing, 2001.

[7] M. V. Arena, O. Younossi, K. Brancato, I. Blickstein, and C. A. Grammich, *Why Has the Cost of Fixed-Wing Aircraft Risen? A Macroscopic Examination of the Trends in U.S. Military Aircraft Costs over the Past Several Decades*. Santa Monica, CA: RAND Corporation, 2008.

[8] F. Austin, G. Carbone, M. Falco, H. Hinz, and M. Lewis, in, ser. Guidance, Navigation, and Control and Co-located Conferences. American Institute of Aeronautics and Astronautics, 1987, ch. Automated maneuvering decisions for air-to-air combat.

[9] F. Austin, G. Carbone, H. Hinz, M. Lewis, and M. Falco, "Game theory for automated maneuvering during air-to-air combat," *Journal of Guidance, Control, and Dynamics*, vol. 13, no. 6, pp. 1143–1149, 1990.

[10] M. Babaeizadeh, I. Frosio, S. Tyree, J. Clemons, and J. Kautz, "Reinforcement learning through asynchronous advantage actor-critic on a gpu," *arXiv preprint arXiv:1611.06256*, 2016.

[11] B. Baker, I. Kanitscheider, T. Markov, Y. Wu, G. Powell, B. McGrew, and I. Mordatch, *Emergent tool use from multi-agent autocurricula*, 2020. arXiv: 1909.07528 [cs.LG].

[12] R. E. Ball, *Fundamentals of Aircraft Combat Survivability Analysis and Design*, 2nd ed., J. A. Schetz, Ed., ser. Education. American Institute of Aeronautics and Astronautics, 1985.

[13] N. Barhate, *Minimal pytorch implementation of proximal policy optimization*, `https://github.com/nikhilbarhate99/PPO-PyTorch`, 2021.

[14] A. Barredo Arrieta, N. Díaz-Rodríguez, J. Del Ser, A. Bennetot, S. Tabik, A. Barbado, S. Garcia, S. Gil-Lopez, D. Molina, R. Benjamins, R. Chatila, and F. Herrera, "Explainable artificial intelligence (xai): Concepts, taxonomies, opportunities and challenges toward responsible ai," *Information Fusion*, vol. 58, pp. 82–115, 2020.

[15] A. Berger, "Beyond blue 4: The past and future transformation of red flag," 2004.

[16] P. Biltgen, "A methodology for capability-based technology evaluation for systems-of-systems," Doctoral dissertation, Georgia Institute of Technology, May 2007.

[17] T. T. Binh and U. Korn, "Mobes: A multiobjective evolution strategy for constrained optimization problems," in *The third international conference on genetic algorithms (Mendel 97)*, Citeseer, vol. 25, 1997, p. 27.

[18] E. Bjorkman, "Have gun, will dogfight," *Air & Space Magazine*, 2015.

[19] M. Blair, "Evolution of the F-86," in *The evolution of aircraft wing design; Proceedings of the Symposium*, 1980, p. 3039.

[20] F. G. Blanchet, P. Legendre, and D. Borcard, "Forward selection of explanatory variables," *Ecology*, vol. 89, no. 9, pp. 2623–2632, 2008.

[21] E. Bonabeau, "Agent-based modeling: Methods and techniques for simulating human systems," *Proceedings of the National Academy of Sciences of the United States of America*, vol. 99, no. 10, pp. 7280–7287, 2002.

[22] A. Borshchev and A. Filippov, "From system dynamics and discrete event to practical agent based modeling : Reasons , techniques , tools," in *Proceedings of the 22nd International Conference of the System Dynamics Society*, Oxford, England, 2004.

[23] A. Bressan. (2010). Noncooperative Differential Games. A Tutorial.

[24] C. Bright, D. Morgan, B. White, S. Krycinsky, J. Rosebrock, S. Wozniak, P. McChesney, B. Smith, and W. H. Mason, *The Hedgehod: A Homeland Defense Interceptor, 2005-2006 AIAA Undergraduate Team Aircraft Design Competition*, online at `http://www.dept.aoe.vt.edu/~mason/Mason_f/VTechT4Hedgehog.pdf`, Virginia Polytechnic Institute, 2005.

[25] W. L. Brogan, *Modern Control Theory*, 2nd ed. Englewood Cliffs, NJ: Prentice-Hall, Inc., 1985.

[26] *Capabilities-Based Assessment (CBA) User's Guide*, 3rd ed., United States Joint Chiefs of Staff, Force Structure, Resources, and Assessments Directorate, Mar. 2009.

[27] W. Chappelle, K. McDonald, and K. McMillan, "Important and critical psychological attributes of USAF MQ-1 predator and MQ-9 reaper pilots according to subject matter experts," SCHOOL OF AEROSPACE MEDICINE WRIGHT PATTERSON AFB OH AEROSPACE MEDICINE DEPT . . ., Tech. Rep., 2011.

[28] P. Clive, J. A. Johnson, M. J. Moss, J. M. Zeh, B. M. Birkmire, and D. D. Hodson, "Advanced Framework for Simulation, Integration and Modeling (AFSIM) (Case Number: 88ABW-2015-2258)," in *Proceedings of the International Conference on Scientific Computing (CSC)*, 2015.

[29] C. D. Connors, "Agent-based modeling methodology for analyzing weapons systems," Theses and Dissertations, Air Force Institute of Technology, 2015.

[30] R. C. Coulter, "Implementation of the pure pursuit path tracking algorithm," Carnegie Mellon University, Pittsburgh, PA, Tech. Rep. CMU-RI-TR-92-01, 1992.

[31] E. Cramer, J. Dennis Jr., P. Frank, R. Lewis, and G. Shubin, "Problem formulation for multidisciplinary optimization," *SIAM Journal on Optimization*, vol. 4, no. 4, pp. 754–776, 1994. eprint: `https://doi.org/10.1137/0804044`.

[32] B. C. Csáji *et al.*, "Approximation with artificial neural networks," *Faculty of Sciences, Etvs Lornd University, Hungary*, vol. 24, no. 48, p. 7, 2001.

[33] P. Cunningham, M. Cord, and S. J. Delany, "Supervised learning," in *Machine learning techniques for multimedia*, Springer, 2008, pp. 21–49.

[34] S. Davies, *Red eagles: America's secret MiGs*. Bloomsbury Publishing, 2011.

[35] K. Deb, A. Pratap, S. Agarwal, and T. Meyarivan, "A fast and elitist multiobjective genetic algorithm: Nsga-ii," *IEEE transactions on evolutionary computation*, vol. 6, no. 2, pp. 182–197, 2002.

[36] Defense Advanced Research Projects Agency, *Explainable Artificial Intelligence (XAI)*, Online at `https://www.darpa.mil/program/explainable-artificial-intelligence`, Aug. 10, 2016.

[37] S. Dowling. (2015). How not to land a fighter jet, (visited on 06/07/2021).

[38] L. A. Dugatkin, *What is "behavior" anyway?* Psychology Today, 2012.

[39] Eckstein, Megan. (2016). F-35B Tactics Evolving As Pilots' Understanding Of Technology Matures, United States Naval Institute, (visited on 03/13/2019).

[40] "Experiment.". (2021). Merriam-Webster.com, (visited on 04/07/2021).

[41] "Feasible.". (2021). Merriam-Webster.com, (visited on 05/06/2021).

[42] ——, (2021). The Oxford English Dictionary, (visited on 05/06/2021).

[43] S. A. Fino, "All the missiles work: Technological dislocations and military innovation," Air University Press, Tech. Rep., 2015, pp. 73–94.

[44] J. W. Forrester, "System dynamics, systems thinking, and soft OR," *System Dynamics Review*, vol. 10, no. 2-3, pp. 245–256, eprint: `https://onlinelibrary.wiley.com/doi/pdf/10.1002/sdr.4260100211`.

[45] ——, "Industrial dynamics: A major breakthrough for decision makers," *Harvard Business Review*, pp. 37–66, 36 1958.

[46] ——, "Industrial dynamics-after the first decade," *Management Science*, vol. 14, no. 7, pp. 398–415, 1968.

[47] S. Friedman, "Air Force: Red Flag cut will save $3.5 million," *Fairbanks Daily News-Miner*, Apr. 2, 2013.

[48] R. D. Gabbert and G. B. Streets, "A comparative analysis of usaf fixed wing aircraft losses in southeast asia combat," Untied States Air Force Flight Dynamics Laboratory, Tech. Rep., 1977.

[49] Z. Ghahramani, "Unsupervised learning," in *Summer School on Machine Learning*, Springer, 2003, pp. 72–112.

[50] A. R. Girard and P. T. Kabamba, "Proportional navigation: Optimal homing and optimal evasion," *SIAM Review*, vol. 57, no. 4, pp. 611–624, 2015.

[51] D. L. Goldfein, *Joint Capabilities Integration and Development System*, 2015.

[52] A. A. Goldstein, "On steepest descent," *Journal of the Society for Industrial and Applied Mathematics, Series A: Control*, vol. 3, no. 1, pp. 147–151, 1965.

[53] K. W. Goodson and A. G. Few Jr, "Effect of leading-edge chord-extensions on subsonic and transonic aerodynamic characteristics of three models having 45 degrees sweptback wings of aspect ratio 4," 1953.

[54] G. Gordon, "A general purpose systems simulation program," in *Proceedings of the Eastern Joint Computer Conference*, IBM Corporation, 1961.

[55] S. E. Gordon, "Rams: Rapid agent-based method for surrogate model development," Doctoral dissertation, Georgia Institute of Technology, 2018.

[56] R. Grant, *The radar game*, 2010.

[57] V. Grimm, U. Berger, F. Bastiansen, S. Eliassen, V. Ginot, J. Giske, J. Goss-Custard, T. Grand, S. K. Heinz, G. Huse, A. Huth, J. U. Jepsen, C. Jørgensen, W. M. Mooij, B. Müller, G. Pe'er, C. Piou, S. F. Railsback, A. M. Robbins, M. M. Robbins, E. Rossmanith, N. Rüger, E. Strand, S. Souissi, R. A. Stillman, R. Vabø, U. Visser, and D. L. DeAngelis, "A standard protocol for describing individual-based and agent-based models," *Ecological Modelling*, vol. 198, no. 1, pp. 115 –126, 2006.

[58] V. Grimm, U. Berger, D. L. DeAngelis, J. G. Polhill, J. Giske, and S. F. Railsback, "The ODD protocol: A review and first update," *Ecological Modelling*, vol. 221, no. 23, pp. 2760 –2768, 2010.

[59] V. Grimm and S. F. Railsback, *Individual-based Modeling and Ecology:* STU - Student edition. Princeton University Press, 2005, ISBN: 9780691096667.

[60] M. H. Hassoun *et al.*, *Fundamentals of artificial neural networks*. MIT press, 1995.

[61] D. M. Hawkins, "The problem of overfitting," *Journal of Chemical Information and Computer Sciences*, vol. 44, no. 1, pp. 1–12, 2004.

[62] Y. Ho, A. Bryson, and S. Baron, "Differential games and optimal pursuit-evasion strategies," *IEEE Transactions on Automatic Control*, vol. 10, no. 4, pp. 385–389, 1965.

[63] R. O. Hundley, "Past revolutions, future transformations. what can the history of revolutions in military affairs tell us about transforming the us military?" RAND CORP SANTA MONICA CA, Tech. Rep., 1999.

[64] C. Hwang and K. Yoon, *Multiple Attribute Decision Making: Methods and Applications : a State-of-the-art Survey*, ser. Lecture notes in economics and mathematical systems. Springer-Verlag, 1981, ISBN: 9783540105589.

[65] R. L. Iman, "Latin hypercube sampling," *Encyclopedia of Quantitative Risk Analysis and Assessment*, 2008.

[66] R. J. James, "A history of radar," *IEE Review*, vol. 35, no. 9, pp. 343–349, 1989.

[67] L. P. Kaelbling, M. L. Littman, and A. W. Moore, "Reinforcement learning: A survey," *Journal of artificial intelligence research*, vol. 4, pp. 237–285, 1996.

[68] E. Kang, E. Jackson, and W. Schulte, "An approach for effective design space exploration," in *Monterey Workshop*, Springer, 2010, pp. 33–54.

[69] K. Kernstine, "Design Spaced Exploration for Stochastic System-of-Systems Simulations Using Adaptive Sequential Experiments," Doctoral dissertation, Georgia Institute of Technology, 2012.

[70] M. R. Kirby, "A Methodology for Technology Identification, Evaluation, and Selection in Conceptual and Preliminary Aircraft Design," Doctoral thesis, Georgia Institute of Technology, 2001.

[71] D. E. Kirk, *Optimal control theory : an introduction*. Mineola, N.Y. : Dover Publications, 2004, Originally published: Englewood Cliffs, N.J. : Prentice-Hall, 1970 (Prentice-Hall networks series), ISBN: 9780486135076.

[72] S. Kullback and R. A. Leibler, "On information and sufficiency," *The Annals of Mathematical Statistics*, vol. 22, no. 1, pp. 79–86, 1951.

[73] S. Legg and M. Hutter, "A formal measure of machine intelligence," *CoRR*, vol. abs/cs/0605024, 2006. arXiv: cs/0605024.

[74] T. P. Lillicrap, J. J. Hunt, A. Pritzel, N. Heess, T. Erez, Y. Tassa, D. Silver, and D. Wierstra, *Continuous control with deep reinforcement learning*, 2019. arXiv: 1509.02971 [cs.LG].

[75] A. Locatelli, *Optimal Control: An Introduction*, 1st ed. Birkhäuser Basel, 2001, ISBN: 978-3-7643-6408-3.

[76] J. W. Locke, "Air superiority at red flag: Mass, technology, and winning the next war," AIR UNIV MAXWELL AFB AL AIR FORCE RESEARCH INST, Tech. Rep., 2009.

[77] Lockheed Martin Corporation. (2020). 5th Generation Capabilities.

[78] C. Macal and M. North, "Tutorial on agent-based modeling and simulation," in *Proceedings of the Winter Simulation Conference, 2005.*, 2005, 14 pp.–.

[79] ——, "Introductory tutorial: Agent-based modeling and simulation," in *Proceedings of the Winter Simulation Conference 2014*, 2014, pp. 6–20.

[80] J. R. Martins and A. B. Lambe, "Multidisciplinary design optimization: A survey of architectures," *AIAA journal*, vol. 51, no. 9, pp. 2049–2075, 2013.

[81] J. D. Mattingly, W. H. Heiser, and D. T. Pratt, *Aircraft engine design*. American Institute of Aeronautics and Astronautics, 2002.

[82] D. N. Mavris, D. A. DeLaurentis, O. Bandte, and M. A. Hale, "A stochastic approach to multi-disciplinary aircraft analysis and design," in *36th AIAA Aerospace Sciences Meeting and Exhibit*, 1998.

[83] S. A. McChrystal, *Joint Capabilities Integration and Development System*, 2009.

[84] S. L. McFarland, *A Concise History of the US Air Force*. Department of the Air Force, 1997.

[85] V. Mnih, K. Kavukcuoglu, D. Silver, A. Graves, I. Antonoglou, D. Wierstra, and M. Riedmiller, "Playing atari with deep reinforcement learning," *arXiv preprint arXiv:1312.5602*, 2013.

[86] Montes, Alexandre. (2019). Red Flag 21-2 creates agile, multi-domain problem-solvers, (visited on 04/07/2021).

[87] B. Müller, F. Bohn, G. Dreßler, J. Groeneveld, C. Klassert, R. Martin, M. Schlüter, J. Schulze, H. Weise, and N. Schwarz, "Describing human decisions in agent-based models – odd + d, an extension of the odd protocol," *Environmental Modelling & Software*, vol. 48, pp. 37 –48, 2013.

[88] S. A. Murtaugh and H. E. Criel, "Fundamentals of proportional navigation," *IEEE Spectrum*, vol. 3, no. 12, pp. 75–85, 1966.

[89] T. Nam, "A generalized sizing method for revolutionary concepts under probabilistic design constraints," PhD, Georgia Institute of Technology, 2007.

[90] J. Nash, "Non-Cooperative Games," Doctoral dissertation, Princeton University, 1950.

[91] National Research Council, *Making the Soldier Decisive on Future Battlefields*. Washington, DC: The National Academies Press, 2013, ISBN: 978-0-309-28453-0.

[92] NATO Warfare Development Command, *Concept Development and Experimentation, A Concept Developer's Toolbox*, version 2.01, 2021, online.

[93] J. Neufeld and G. M. Watson Jr, "Coalition Air Warfare in the Korean War 1950-1953," OFFICE OF AIR FORCE HISTORY WASHINGTON DC, Tech. Rep., 2005.

[94] J. von Neumann and O. Morgenstern, *Theory of Games and Economic Behavior*. Princeton University Press, 1944.

[95] S. O. Nichols, "21st century air-to-air short range weapon requirements," AIR COMMAND and STAFF COLL MAXWELL AFB AL, Tech. Rep., 1998.

[96] J. Nocedal and S. Wright, *Numerical optimization*. Springer Science & Business Media, 2006.

[97] Office of Aerospace Studies, *Analysis of Alternatives (AoA) Handbook, A practical guide to analyses of alternatives*, Air Force Materiel Command, 2008.

[98] ——, *Analysis of Alternatives (AoA) Handbook, A practical guide to analyses of alternatives*, Air Force Materiel Command, 2016.

[99] Office of the Comptroller General, "C-5A Wing Modification: A Case Study Illustrating Problems In The Defense Weapons Acquisition Process," PLRD-82-38, United States Government Accountability Office, 1982.

[100] M. O'Hanlon, "Forecasting change in military technology, 2020-2040," *Military Technology*, vol. 2020, p. 2040, 2018.

[101] "Organism.". (2019). Merriam-Webster.com, (visited on 02/14/2019).

[102] A. Paszke, S. Gross, S. Chintala, G. Chanan, E. Yang, Z. DeVito, Z. Lin, A. Desmaison, L. Antiga, and A. Lerer, "Automatic differentiation in pytorch," 2017.

[103] A. Paszke, S. Gross, F. Massa, A. Lerer, J. Bradbury, G. Chanan, T. Killeen, Z. Lin, N. Gimelshein, L. Antiga, A. Desmaison, A. Kopf, E. Yang, Z. DeVito, M. Raison, A. Tejani, S. Chilamkurthy, B. Steiner, L. Fang, J. Bai, and S. Chintala, "Pytorch: An imperative style, high-performance deep learning library," in *Advances in Neural Information Processing Systems 32*, H. Wallach, H. Larochelle, A. Beygelzimer, F. d'Alché Buc, E. Fox, and R. Garnett, Eds., Curran Associates, Inc., 2019, pp. 8024–8035.

[104] R. E. van Patten. (1991). G-Locked and the Fighter Jock, (visited on 06/07/2021).

[105] A. P. Pope, J. S. Ide, D. Micovic, H. Diaz, D. Rosenbluth, L. Ritholtz, J. C. Twedt, T. T. Walker, K. Alcedo, and D. Javorsek, *Hierarchical reinforcement learning for air-to-air combat*, 2021. arXiv: 2105.00990 [cs.LG].

[106] "Principle.". (2019). The Oxford English Dictionary, (visited on 02/14/2019).

[107] R. V. Rao, "Introduction to multiple attribute decision-making (madm) methods," in *Decision Making in the Manufacturing Environment: Using Graph Theory and Fuzzy Multiple Attribute Decision Making Methods*. London: Springer London, 2007, pp. 27–41, ISBN: 978-1-84628-819-7.

[108] S. S. Rao, *Engineering Optimization: Theory and Practice*. New Age International, 2000, ISBN: 9788122411492.

[109] D. W. Rhyne, "Acquisition Program Management Challenges in Afghanistan. Part 1: Requirements Generation," Defense Acquisition University Fort Belvoir United States, Tech. Rep., 2011.

[110] P. Richmond, "Resolving conflicts between multiple competing agents in parallel simulations," in *Euro-Par 2014: Parallel Processing Workshops*, L. Lopes, J. Žilinskas, A. Costan, R. G. Cascella, G. Kecskemeti, E. Jeannot, M. Cannataro, L. Ricci, S. Benkner, S. Petit, V. Scarano, J. Gracia, S. Hunold, S. L. Scott, S. Lankes, C. Lengauer, J. Carretero, J. Breitbart, and M. Alexander, Eds., Cham: Springer International Publishing, 2014, pp. 383–394, ISBN: 978-3-319-14325-5.

[111] D. Ross, "Game theory," in *The Stanford Encyclopedia of Philosophy*, E. N. Zalta, Ed., Spring 2019, Metaphysics Research Lab, Stanford University, 2019.

[112] J. Rust, "Using randomization to break the curse of dimensionality," *Econometrica: Journal of the Econometric Society*, pp. 487–516, 1997.

[113] S. Safavian and D. Landgrebe, "A survey of decision tree classifier methodology," *IEEE Transactions on Systems, Man, and Cybernetics*, vol. 21, no. 3, pp. 660–674, 1991.

[114] B. Sauser, J. E. Ramirez-Marquez, R. Magnaye, and W. Tan, "A systems approach to expanding the technology readiness level within defense acquisition," STEVENS INST OF TECH HOBOKEN NJ SCHOOL OF SYSTEMS and ENTERPRISES, Tech. Rep., 2009.

[115] J. Schlight, *A War Too Long: The USAF in Southeast Asia 1961-1975*. University Press of the Pacific, 1996.

[116] D. Schrage and D. Mavris, "Technology for affordability – how to define, measure, evaluate, and implement it?" In *50th National Forum of the American Helicopter Society*, 1994.

[117] D. Schrage, K. Taggart, and D. DeLaurentis, "IPPD concept development process for future combat system," in *9th AIAA/ISSMO Symposium on Multidisciplinary Analysis and Optimization*, 2002, p. 5619.

[118] J. Schulman, S. Levine, P. Moritz, M. I. Jordan, and P. Abbeel, "Trust region policy optimization," *CoRR*, vol. abs/1502.05477, 2015. arXiv: 1502.05477.

[119] ——, *Trust region policy optimization*, 2017. arXiv: 1502.05477 [cs.LG].

[120] J. Schulman, F. Wolski, P. Dhariwal, A. Radford, and O. Klimov, *Proximal policy optimization algorithms*, 2017. arXiv: 1707.06347 [cs.LG].

[121] P. Scott, "Birth of the jet engine," *Mechanical Engineering*, vol. 117, no. 1, p. 66, 1995.

[122] I. G. Shaw, "Predator Empire: The Geopolitics of US Drone Warfare," *Geopolitics*, vol. 18, no. 3, pp. 536–559, 2013.

[123] R. Shaw, *Fighter Combat Tactics and Maneuvering*. Annapolis, MD: Naval Institutes Press, 1985.

[124] Y. Shi and R. Eberhart, "A modified particle swarm optimizer," in *1998 IEEE international conference on evolutionary computation proceedings. IEEE world congress on computational intelligence (Cat. No. 98TH8360)*, IEEE, 1998, pp. 69–73.

[125] J. J. Siegner, "Analysis of alternatives: Multivariate consideration.," AIR FORCE INSTITUTE OF TECHNOLOGY, WRIGHT-PATTERSON AIR FORCE BASE, OHIO, SCHOOL OF ENGINEERING, Tech. Rep., 1998.

[126] D. Silver, G. Lever, N. Heess, T. Degris, D. Wierstra, and M. Riedmiller, "Deterministic policy gradient algorithms," in *International conference on machine learning*, PMLR, 2014, pp. 387–395.

[127] D. Silver, J. Schrittwieser, K. Simonyan, I. Antonoglou, A. Huang, A. Guez, T. Hubert, L. Baker, M. Lai, A. Bolton, Y. Chen, T. Lillicrap, F. Hui, L. Sifre, G. van den Driessche, T. Graepel, and D. Hassabis, "Mastering the game of go without human knowledge," *Nature*, vol. 550, pp. 354 –359, 2017, Article.

[128] Z. Siyu, W. Wenhai, Z. Shengming, and Q. Zhigang, "A new situation assessment model for modern within-visual-range air combat," *Procedia Engineering*, vol. 29, pp. 339–343, 2012, 2012 International Workshop on Information and Electronics Engineering.

[129] B. F. Skinner, "The generic nature of the concepts of stimulus and response," *Journal of General Psychology*, pp. 40–63, 1935.

[130] ——, *The behavior of organisms: An experimental analysis*. Appleton-Century Company, Inc., 1938.

[131] L. N. Smith, "A disciplined approach to neural network hyper-parameters: Part 1– learning rate, batch size, momentum, and weight decay," *arXiv preprint arXiv:1803.09820*, 2018.

[132] Smithsonian Institution. (). UAV, General Atomics MQ-1L Predator A, National Air and Space Museum, (visited on 08/14/2021).

[133] L Spearman, "Some fighter aircraft trends," 1985.

[134] M. L. Spearman, "Historical development of worldwide guided missiles," 1983.

[135] P. Sprey, "Comparing the Effectiveness of Air-to-Air Fighters: F-86 to F-18," Pierre M. Sprey, Inc., 1982.

[136] J. Stillion, "Trends in Air-to-Air Combat, Implications for Future Air Superiority," Center for Strategic and Budgetary Assessments, 2015.

[137] R. S. Sutton, "Temporal credit assignment in reinforcement learning," Doctoral dissertation, University of Massachusetts, Amherst, 1984.

[138] R. S. Sutton and A. G. Barto, *Reinforcement Learning, An Introduction*, 2nd ed. Cambridge, MA: MIT Press, 2018.

[139] R. S. Sutton, D. A. McAllester, S. P. Singh, Y. Mansour, *et al.*, "Policy gradient methods for reinforcement learning with function approximation.," in *NIPs*, Citeseer, vol. 99, 1999, pp. 1057–1063.

[140] Y. Tadjdeh. (2021). SOCOM Keeps Pushing for a New 'Armed Overwatch' Aircraft, (visited on 06/05/2021).

[141] S. A. Tangen, "A methodology for the quantification of doctrine and materiel approaches in a capability-based assessment," Doctoral dissertation, Georgia Institute of Technology, 2009.

[142] J. S. Thach, "Butch O'Hare and the Thach Weave," *Naval History Magazine*, vol. 6, no. 1, 1992.

[143] The European Space Agency. (n.d.). How many stars are there in the universe? (Visited on 05/18/2021).

[144] J. A. Tirpak. (2021). The Raiders Comes Out of the Black, (visited on 06/05/2021).

[145] J. Tromp and G. Farnebäck, "Combinatorics of go," in *Computers and Games*, H. J. van den Herik, P. Ciancarini, and H. H. L. M. J. Donkers, Eds., Berlin, Heidelberg: Springer Berlin Heidelberg, 2007, pp. 84–99, ISBN: 978-3-540-75538-8.

[146] A. Turner, "A Methodology for the Development of Models for the Simulation of Non-Observable Systems," Doctoral dissertation, Georgia Institute of Technology, 2014.

[147] United States Air Force. (2015). Countering MiGs: Air-to-Air Combat Over North Vietnam.

[148] ——, (2015). MQ-1B Predator.

[149] ——, (2019). 414th Combat Training Squadron "Red Flag", (visited on 03/06/2019).

[150] United States Navy. (Feb. 2021). F/A-18A-D Hornet and F/A-18E-F Super Hornet Strike Fighter.

[151] U.S. Department of Homeland Security, *Verification, Validation and Acreditation (VV&A) of Models and Simulations (M&S)*, 2006.

[152] USAF Historical Division, "United States Air Force Operations In The Korean Conflict," Research Studies Institute, Air University, Department of the Air Force, Tech. Rep., 1956.

[153] O. Vinyals, I. Babuschkin, W. M. Czarnecki, M. Mathieu, A. Dudzik, J. Chung, D. H. Choi, R. Powell, T. Ewalds, P. Georgiev, J. Oh, D. Horgan, M. Kroiss, I. Danihelka, A. Huang, L. Sifre, T. Cai, J. P. Agapiou, M. Jaderberg, A. S. Vezhnevets, R. Leblond, T. Pohlen, V. Dalibard, D. Budden, Y. Sulsky, J. Molloy, T. L. Paine, C. Gulcehre, Z. Wang, T. Pfaff, Y. Wu, R. Ring, D. Yogatama, D. Wünsch, K. McKinney, O. Smith, T. Schaul, T. Lillicrap, K. Kavukcuoglu, D. Hassabis, C. Apps, and D. Silver, "Grandmaster level in StarCraft II using multi-agent reinforcement learning," *Nature*, vol. 575, no. 7782, pp. 350–354, 2019.

[154] K. Virtanen, T. Raivio, and R. P. Hamalainen, "Modeling pilot's sequential maneuvering decisions by a multistage influence diagram," *Journal of Guidance, Control, and Dynamics*, vol. 27, no. 4, pp. 665–677, 2004.

[155] S. J. Walker, "Interoperability at the Speed of Sound: Canada-United States Aerospace Cooperation... Modernizing the CF-18 Hornet," QUEEN'S UNIV KINGSTON (ONTARIO) CENTREFOR INTERNATIONAL RELATIONS, Tech. Rep., 2000.

[156] C. Watkins, "Learning from delayed rewards," Doctoral dissertation, King's College, 1989.

[157] D. Whitley, "A genetic algorithm tutorial," *Statistics and computing*, vol. 4, no. 2, pp. 65–85, 1994.

[158] B. M. Wilamowski, "Neural network architectures and learning algorithms," *IEEE Industrial Electronics Magazine*, vol. 3, no. 4, pp. 56–63, 2009.

[159] S. Wilkinson, *The Goldilocks Fighter: The F6F Hellcat*, Online at `http://www.historynet.com/goldilocks-fighter-f6f-hellcat.htm`, 2017.

[160] C. Young and K. Holsteen, "Model uncertainty and robustness: A computational framework for multimodel analysis," *Sociological Methods & Research*, vol. 46, no. 1, pp. 3–40, 2017.

[161] O. Younossi, M. V. Arena, R. M. Moore, M. A. Lorell, J. Mason, and J. C. Graser, *Military Jet Engine Acquisition: Technology Basics and Cost-Estimating Methodology*. Santa Monica, CA: RAND Corporation, 2003.

[162] L. A. Zhang, J. Xu, D. Gold, J. Hagen, A. K. Kochhar, A. J. Lohn, and O. A. Osoba, *Air Dominance Through Machine Learning: A Preliminary Exploration of Artificial Intelligence-Assisted Mission Planning*. Santa Monica, CA: RAND Corporation, 2020.

[163] F. Zwicky, *Discovery, Invention, Research Through the Morphological Approach*. Macmillan, New York, 1969.

VITA

Mackenzie Lau was born in Honolulu, Hawai‘i. He graduated from Punahou High School in Honolulu, Hawai‘i with the Class of 2011. He went on to earn his Bachelor of Science in Mechanical Engineering from the University of Hawai‘i at Mānoa in 2015 before beginning his graduate studies at the Georgia Institute of Technology. He earned his Master of Science in Aerospace Engineering in 2017 under the guidance of Professor Dimitri Mavris at the Aerospace Systems Design Laboratory before beginning the path to his PhD in earnest.